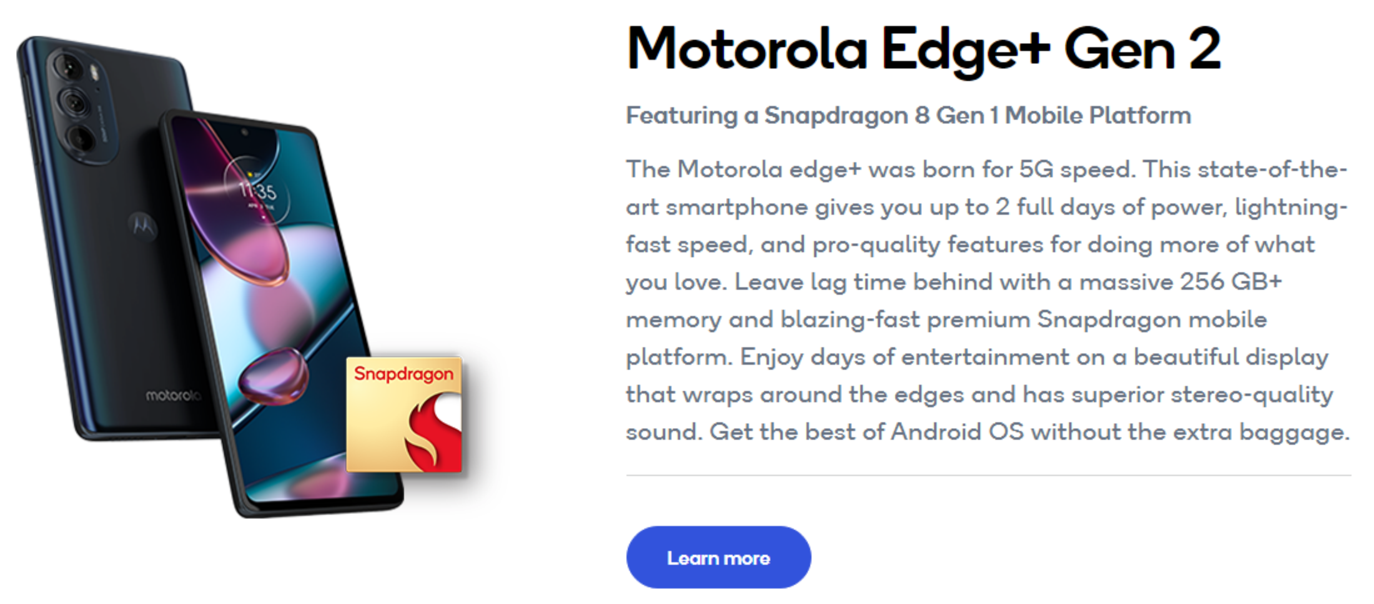
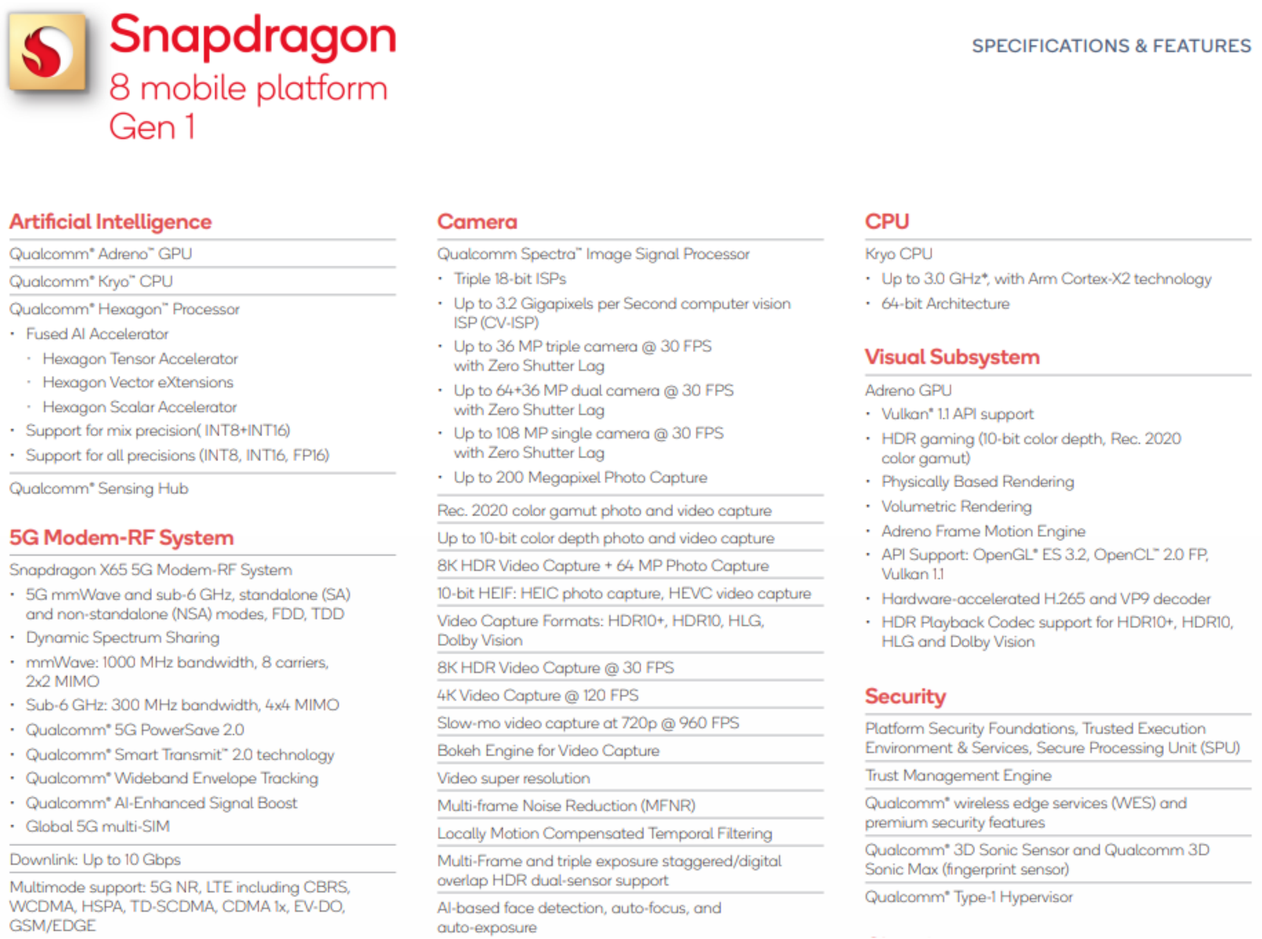
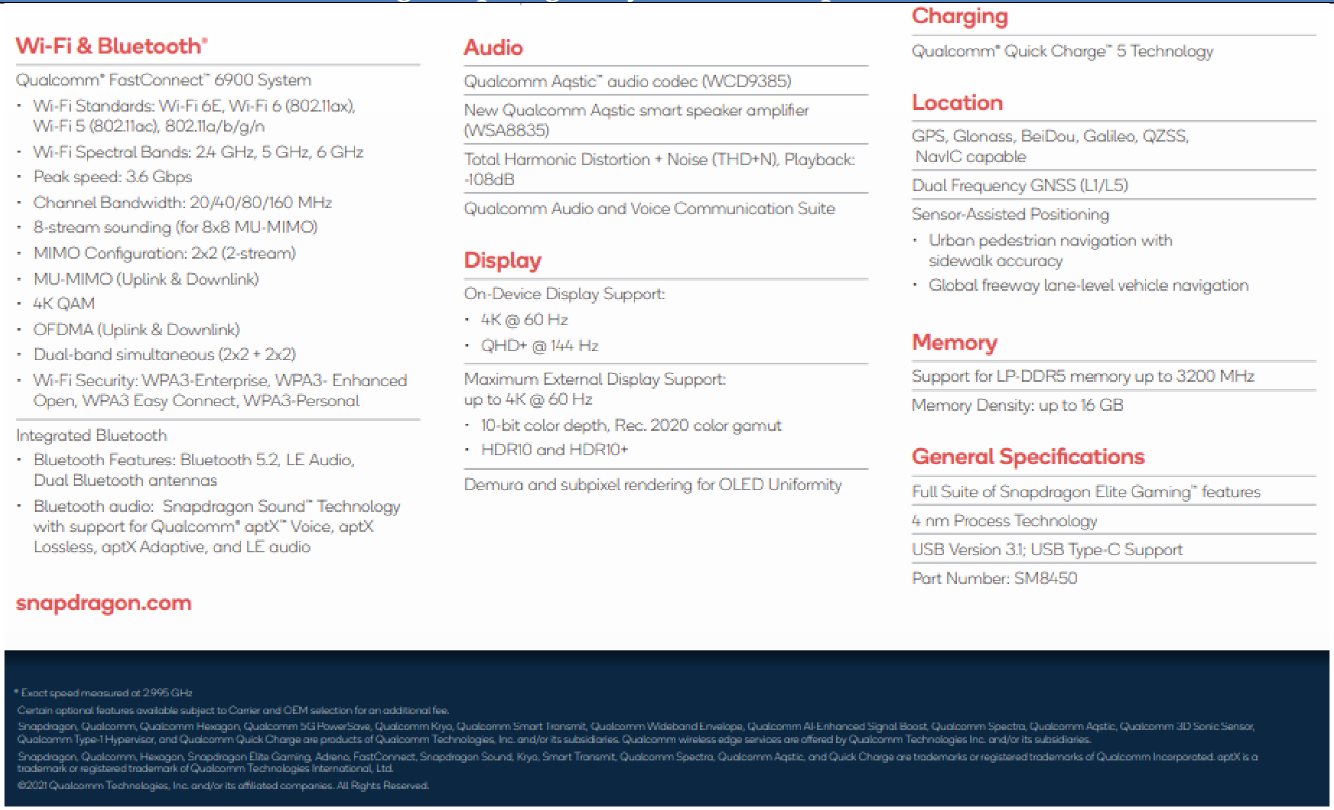
EXHIBIT 033

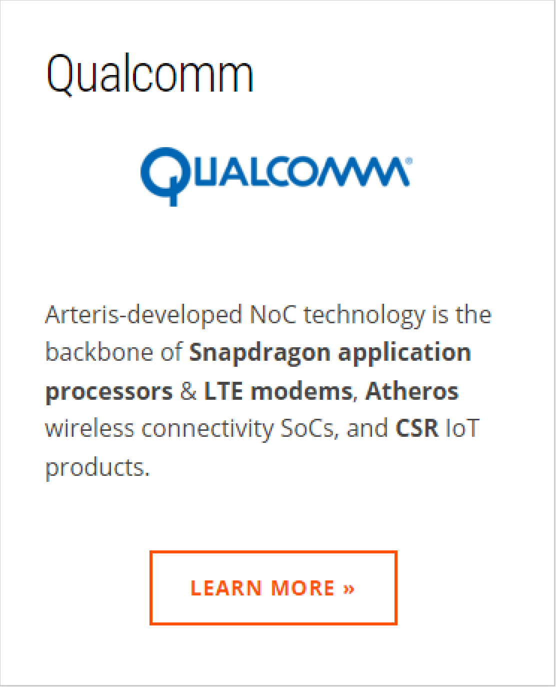
'449 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
10. Method for exchanging messages in an integrated circuit comprising a plurality of modules,	Without conceding that the preamble of claim 10 of the '449 Patent is limiting, the Motorola Edge+ Gen 2 (hereinafter, the "Motorola product") performs a method for exchanging messages in an integrated circuit comprising a plurality of modules, either literally or under the doctrine of equivalents. The Motorola product includes an integrated circuit. For example, the Motorola product includes the Qualcomm Snapdragon 8 Gen 1 Mobile Platform system on chip (hereinafter, the "Snapdragon SoC").  **Motorola Edge+ Gen 2** Featuring a Snapdragon 8 Gen 1 Mobile Platform The Motorola edge+ was born for 5G speed. This state-of-the-art smartphone gives you up to 2 full days of power, lightning-fast speed, and pro-quality features for doing more of what you love. Leave lag time behind with a massive 256 GB+ memory and blazing-fast premium Snapdragon mobile platform. Enjoy days of entertainment on a beautiful display that wraps around the edges and has superior stereo-quality sound. Get the best of Android OS without the extra baggage.

[1] The Motorola product is charted as a representative product made used, sold, offered for sale, and/or imported by Motorola. The citations to evidence contained herein are illustrative and should not be understood to be limiting. The right is expressly reserved to rely upon additional or different evidence, or to rely on additional citations to the evidence cited already cited herein.

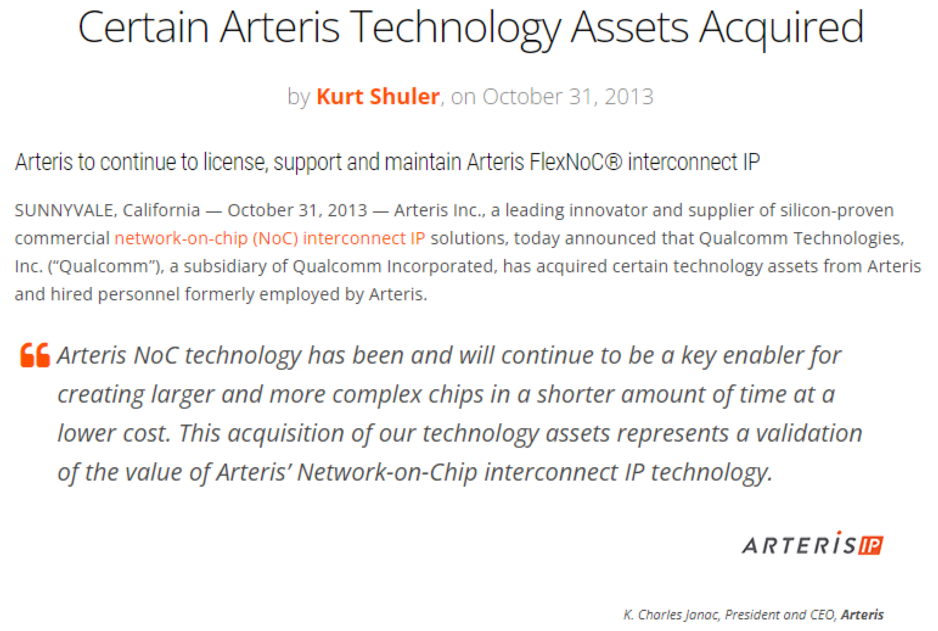
U.S. Patent No. 7,373,449 (Radulescu and Goossens)
"Apparatus and method for communicating in an integrated circuit"

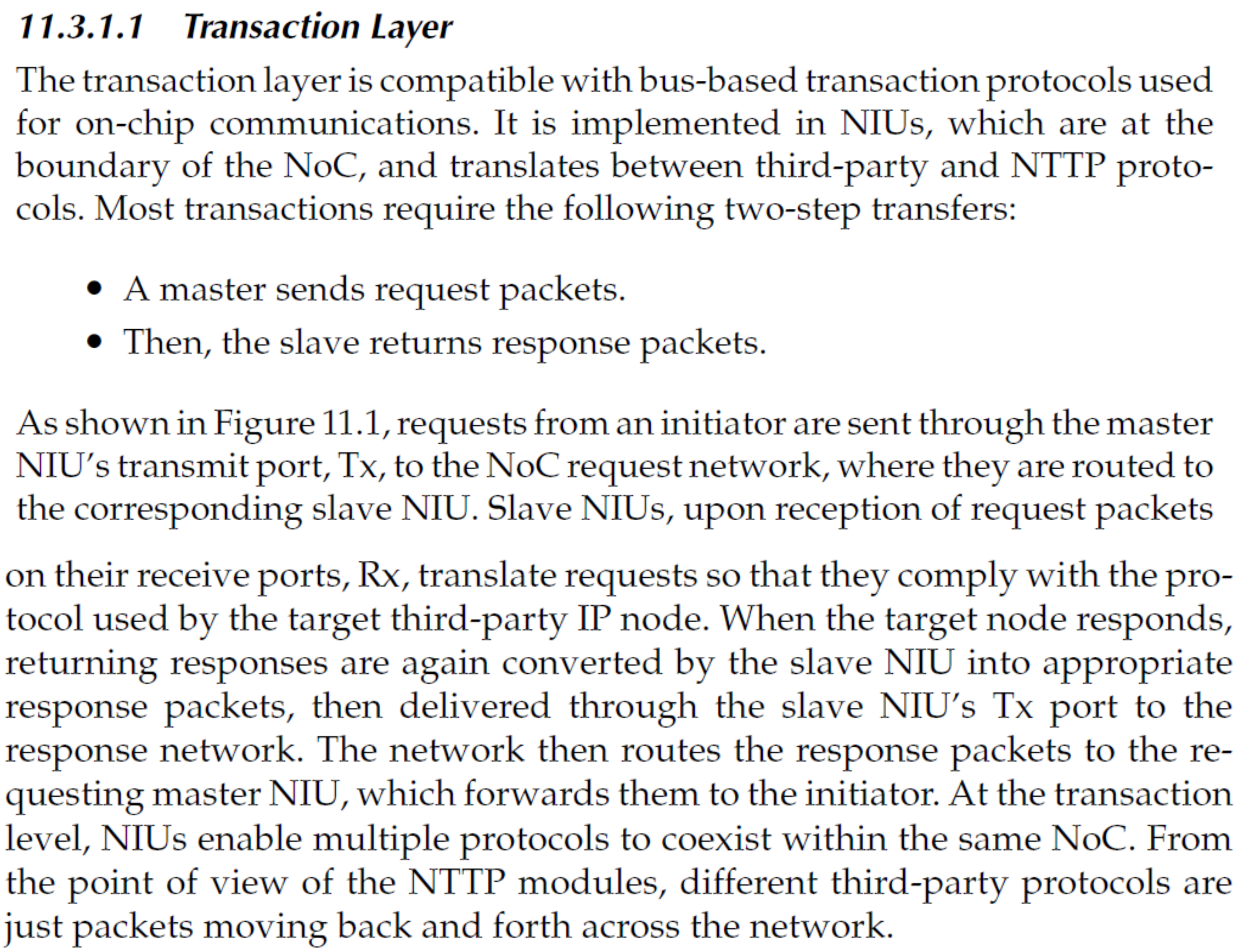
'449 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	https://www.qualcomm.com/snapdragon/device-finder/motorola-edge--gen-2 The Snapdragon SoC comprises a plurality of modules, for example Qualcomm Adreno GPU; Qualcomm Kryo CPU; Qualcomm Hexagon Processor; and Platform Security Foundations, Trusted Execution Environment & Services, Secure Processing Unit (SPU):  Snapdragon 8 mobile platform Gen 1 SPECIFICATIONS & FEATURES **Artificial Intelligence** Qualcomm® Adreno™ GPU Qualcomm® Kryo™ CPU Qualcomm® Hexagon™ Processor • Fused AI Accelerator · Hexagon Tensor Accelerator · Hexagon Vector eXtensions · Hexagon Scalar Accelerator • Support for mix precision(INT8+INT16) • Support for all precisions (INT8, INT16, FP16) Qualcomm® Sensing Hub **5G Modem-RF System** Snapdragon X65 5G Modem-RF System • 5G mmWave and sub-6 GHz, standalone (SA) and non-standalone (NSA) modes, FDD, TDD • Dynamic Spectrum Sharing • mmWave: 1000 MHz bandwidth, 8 carriers, 2x2 MIMO • Sub-6 GHz: 300 MHz bandwidth, 4x4 MIMO • Qualcomm® 5G PowerSave 2.0 • Qualcomm® Smart Transmit™ 2.0 technology • Qualcomm® Wideband Envelope Tracking • Qualcomm® AI-Enhanced Signal Boost • Global 5G multi-SIM Downlink: Up to 10 Gbps Multimode support: 5G NR, LTE including CBRS, WCDMA, HSPA, TD-SCDMA, CDMA 1x, EV-DO, GSM/EDGE **Camera** Qualcomm Spectra™ Image Signal Processor • Triple 18-bit ISPs • Up to 3.2 Gigapixels per Second computer vision ISP (CV-ISP) • Up to 36 MP triple camera @ 30 FPS with Zero Shutter Lag • Up to 64+36 MP dual camera @ 30 FPS with Zero Shutter Lag • Up to 108 MP single camera @ 30 FPS with Zero Shutter Lag • Up to 200 Megapixel Photo Capture Rec. 2020 color gamut photo and video capture Up to 10-bit color depth photo and video capture 8K HDR Video Capture + 64 MP Photo Capture 10-bit HEIF: HEIC photo capture, HEVC video capture Video Capture Formats: HDR10+, HDR10, HLG, Dolby Vision 8K HDR Video Capture @ 30 FPS 4K Video Capture @ 120 FPS Slow-mo video capture at 720p @ 960 FPS Bokeh Engine for Video Capture Video super resolution Multi-frame Noise Reduction (MFNR) Locally Motion Compensated Temporal Filtering Multi-Frame and triple exposure staggered/digital overlap HDR dual-sensor support AI-based face detection, auto-focus, and auto-exposure **CPU** Kryo CPU • Up to 3.0 GHz*, with Arm Cortex-X2 technology • 64-bit Architecture **Visual Subsystem** Adreno GPU • Vulkan® 1.1 API support • HDR gaming (10-bit color depth, Rec. 2020 color gamut) • Physically Based Rendering • Volumetric Rendering • Adreno Frame Motion Engine • API Support: OpenGL® ES 3.2, OpenCL™ 2.0 FP, Vulkan 1.1 • Hardware-accelerated H.265 and VP9 decoder • HDR Playback Codec support for HDR10+, HDR10, HLG and Dolby Vision **Security** Platform Security Foundations, Trusted Execution Environment & Services, Secure Processing Unit (SPU) Trust Management Engine Qualcomm® wireless edge services (WES) and premium security features Qualcomm® 3D Sonic Sensor and Qualcomm 3D Sonic Max (fingerprint sensor) Qualcomm® Type-1 Hypervisor

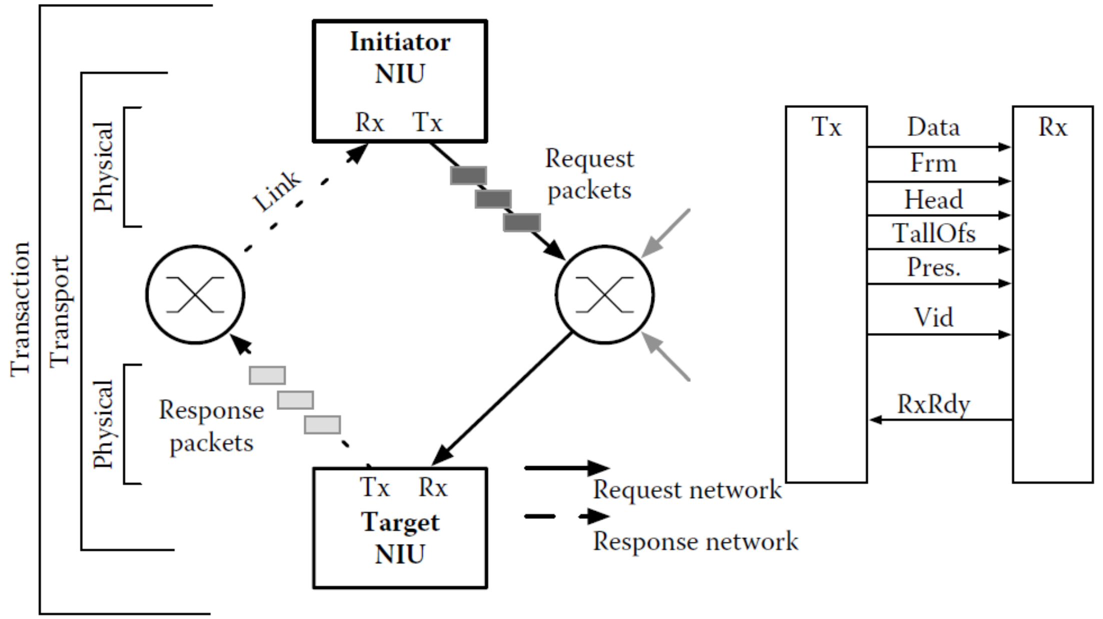
'449 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 **Wi-Fi & Bluetooth®** Qualcomm® FastConnect™ 6900 System • Wi-Fi Standards: Wi-Fi 6E, Wi-Fi 6 (802.11ax), Wi-Fi 5 (802.11ac), 802.11a/b/g/n • Wi-Fi Spectral Bands: 2.4 GHz, 5 GHz, 6 GHz • Peak speed: 3.6 Gbps • Channel Bandwidth: 20/40/80/160 MHz • 8-stream sounding (for 8x8 MU-MIMO) • MIMO Configuration: 2x2 (2-stream) • MU-MIMO (Uplink & Downlink) • 4K QAM • OFDMA (Uplink & Downlink) • Dual-band simultaneous (2x2 + 2x2) • Wi-Fi Security: WPA3-Enterprise, WPA3- Enhanced Open, WPA3 Easy Connect, WPA3-Personal Integrated Bluetooth • Bluetooth Features: Bluetooth 5.2, LE Audio, Dual Bluetooth antennas • Bluetooth audio: Snapdragon Sound™ Technology with support for Qualcomm® aptX™ Voice, aptX Lossless, aptX Adaptive, and LE audio snapdragon.com **Audio** Qualcomm Aqstic™ audio codec (WCD9385) New Qualcomm Aqstic smart speaker amplifier (WSA8835) Total Harmonic Distortion + Noise (THD+N), Playback: -108dB Qualcomm Audio and Voice Communication Suite **Display** On-Device Display Support: • 4K @ 60 Hz • QHD+ @ 144 Hz Maximum External Display Support: up to 4K @ 60 Hz • 10-bit color depth, Rec. 2020 color gamut • HDR10 and HDR10+ Demura and subpixel rendering for OLED Uniformity **Charging** Qualcomm® Quick Charge™ 5 Technology **Location** GPS, Glonass, BeiDou, Galileo, QZSS, NavIC capable Dual Frequency GNSS (L1/L5) Sensor-Assisted Positioning • Urban pedestrian navigation with sidewalk accuracy • Global freeway lane-level vehicle navigation **Memory** Support for LP-DDR5 memory up to 3200 MHz Memory Density: up to 16 GB **General Specifications** Full Suite of Snapdragon Elite Gaming™ features 4 nm Process Technology USB Version 3.1; USB Type-C Support Part Number: SM8450 * Exact speed measured at 2.995 GHz Certain optional features available subject to Carrier and OEM selection for an additional fee. Snapdragon, Qualcomm, Qualcomm Hexagon, Qualcomm 5G PowerSave, Qualcomm Kryo, Qualcomm Smart Transmit, Qualcomm Wideband Envelope, Qualcomm AI-Enhanced Signal Boost, Qualcomm Spectra, Qualcomm Aqstic, Qualcomm 3D Sonic Sensor, Qualcomm Type-1 Hypervisor, and Qualcomm Quick Charge are products of Qualcomm Technologies, Inc. and/or its subsidiaries. Qualcomm wireless edge services are offered by Qualcomm Technologies Inc. and/or its subsidiaries. Snapdragon, Qualcomm, Hexagon, Snapdragon Elite Gaming, Adreno, FastConnect, Snapdragon Sound, Kryo, Smart Transmit, Qualcomm Spectra, Qualcomm Aqstic, and Quick Charge are trademarks or registered trademarks of Qualcomm Incorporated. aptX is a trademark or registered trademark of Qualcomm Technologies International, Ltd. ©2021 Qualcomm Technologies, Inc. and/or its affiliated companies. All Rights Reserved. https://www.qualcomm.com/content/dam/qcomm-martech/dm-assets/documents/snapdragon-8-gen-1-mobile-platform-product-brief.pdf The Snapdragon SoC included in the Motorola product utilizes Arteris network on chip interconnect technology, and/or a derivative thereof, (collectively, the "Arteris NoC") to exchange messages:

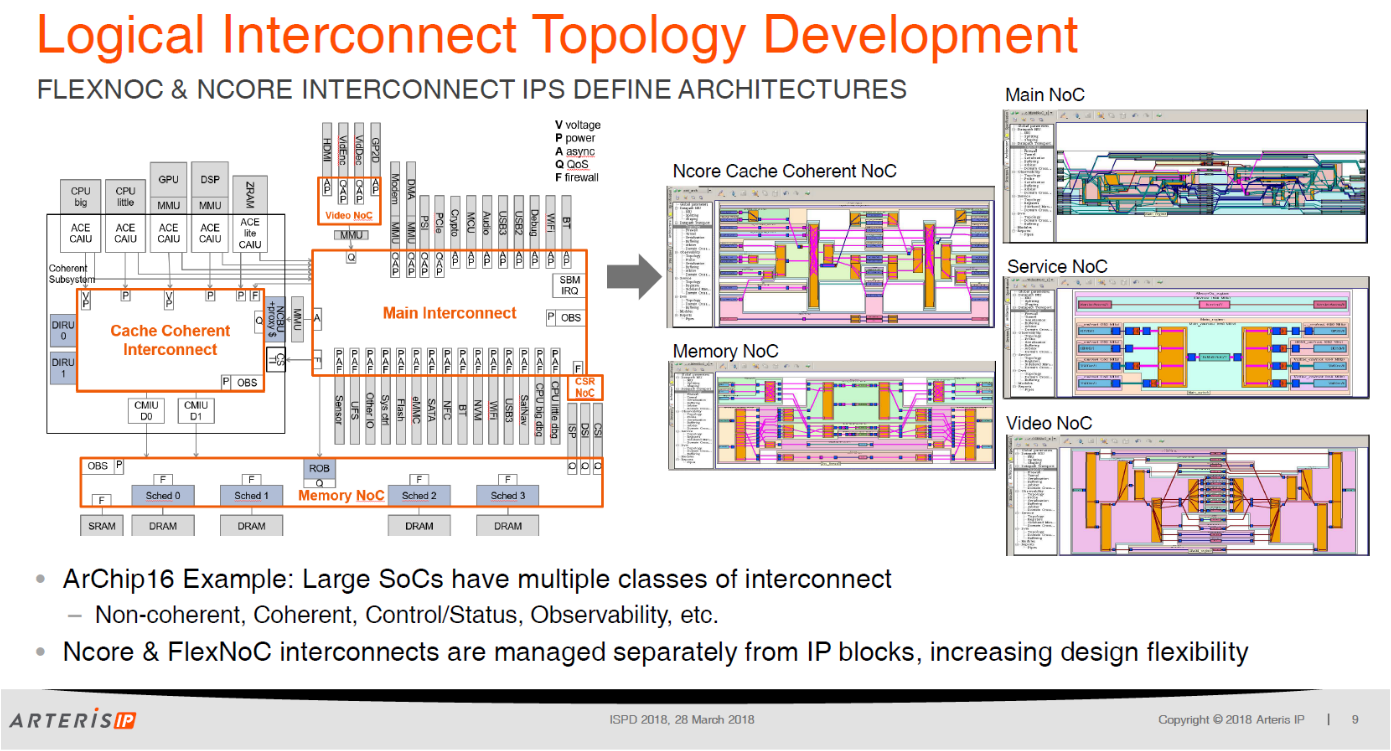
'449 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 Qualcomm Arteris-developed NoC technology is the backbone of **Snapdragon application processors** & **LTE modems**, **Atheros** wireless connectivity SoCs, and **CSR** IoT products. LEARN MORE » https://web.archive.org/web/20210514110614/https://www.arteris.com/customers

U.S. Patent No. 7,373,449 (Radulescu and Goossens)
"Apparatus and method for communicating in an integrated circuit"

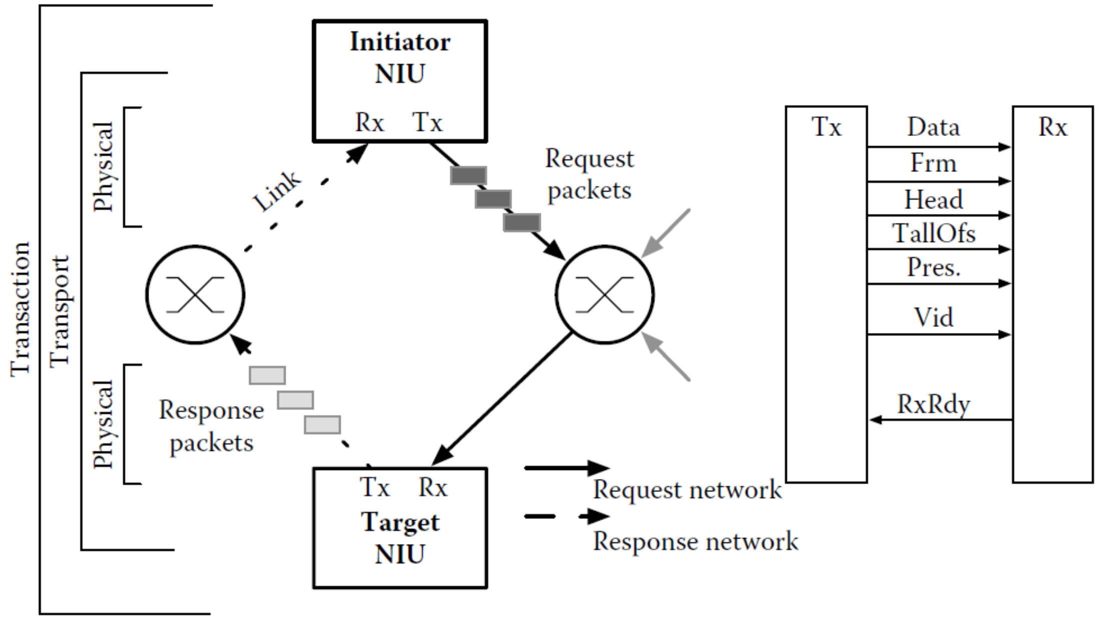
'449 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 Certain Arteris Technology Assets Acquired by **Kurt Shuler**, on October 31, 2013 Arteris to continue to license, support and maintain Arteris FlexNoC® interconnect IP SUNNYVALE, California — October 31, 2013 — Arteris Inc., a leading innovator and supplier of silicon-proven commercial network-on-chip (NoC) interconnect IP solutions, today announced that Qualcomm Technologies, Inc. ("Qualcomm"), a subsidiary of Qualcomm Incorporated, has acquired certain technology assets from Arteris and hired personnel formerly employed by Arteris. *Arteris NoC technology has been and will continue to be a key enabler for creating larger and more complex chips in a shorter amount of time at a lower cost. This acquisition of our technology assets represents a validation of the value of Arteris' Network-on-Chip interconnect IP technology.* ARTERIS IP *K. Charles Janac, President and CEO, **Arteris*** https://www.arteris.com/press-releases/Qualcomm-Arteris-asset-acquisition-2013_oct_31; https://www.fiercewireless.com/tech/qualcomm-acquires-arteris-noc-tech-assets-team The Arteris NoC exchanges messages in the Snapdragon SoC included in the Motorola product. For example, in the Arteris NoC, "[m]ost transactions require the following two-step transfers," including "[a] master send[ing] request packets" and "the slave return[ing] response packets":

'449 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 ***11.3.1.1 Transaction Layer*** The transaction layer is compatible with bus-based transaction protocols used for on-chip communications. It is implemented in NIUs, which are at the boundary of the NoC, and translates between third-party and NTTP protocols. Most transactions require the following two-step transfers: • A master sends request packets. • Then, the slave returns response packets. As shown in Figure 11.1, requests from an initiator are sent through the master NIU's transmit port, Tx, to the NoC request network, where they are routed to the corresponding slave NIU. Slave NIUs, upon reception of request packets on their receive ports, Rx, translate requests so that they comply with the protocol used by the target third-party IP node. When the target node responds, returning responses are again converted by the slave NIU into appropriate response packets, then delivered through the slave NIU's Tx port to the response network. The network then routes the response packets to the requesting master NIU, which forwards them to the initiator. At the transaction level, NIUs enable multiple protocols to coexist within the same NoC. From the point of view of the NTTP modules, different third-party protocols are just packets moving back and forth across the network.

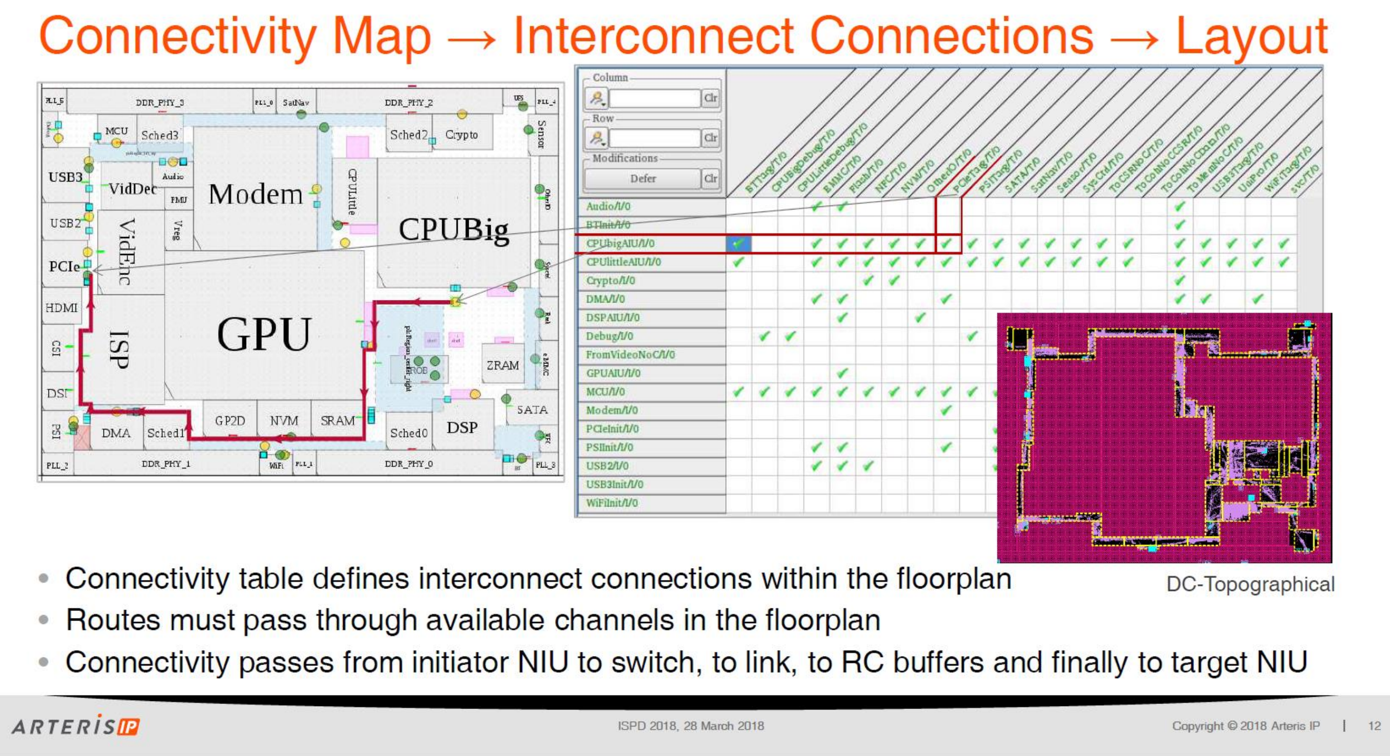
'449 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 Transaction Transport Physical Physical Initiator NIU Rx Tx Link Request packets Response packets Tx Rx Target NIU Request network Response network Tx Data Frm Head TallOfs Pres. Vid RxRdy Rx **FIGURE 11.1** NTTP protocol layers mapped on NoC units and Media Independent NoC Interface—MINI. *See* Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 312-313; *see id* at 308 (explaining that Chapter 11 of this book describes the function of the Arteris NoC: "In this chapter we will present an MPSoC platform […] using Arteris NoC as communication infrastructure.").
the messages between the modules being	Without conceding that the preamble of claim 10 of the '449 Patent is limiting, the Arteris NoC exchanges messages between modules in the Snapdragon SoC included in the Motorola product over connections via a network, wherein said connections comprises a set of communication

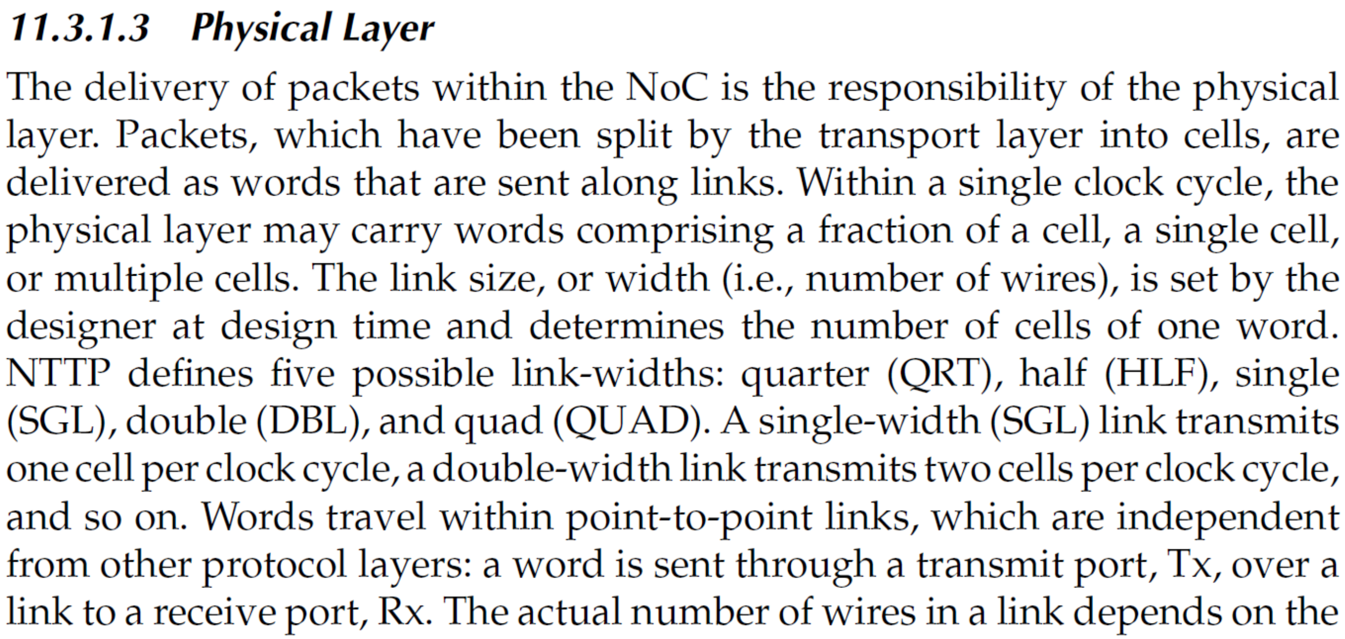
'449 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
exchanged over connections via a network, wherein said connections comprises a set of communication channels each having a set of connection properties, any communication channel being independently configurable,	channels each having a set of connection properties any communication channel being independently configurable, either literally or under the doctrine of equivalents. A large SoC, such as the Snapdragon SoC included in the Motorola product may include multiple classes of Arteris NoC interconnect network:  Logical Interconnect Topology Development FLEXNOC & NCORE INTERCONNECT IPS DEFINE ARCHITECTURES Main NoC Ncore Cache Coherent NoC Service NoC Memory NoC Video NoC V voltage P power A async Q QoS F firewall CPU big CPU little GPU DSP MMU ZRAM ACE CAIU ACE lite CAIU Coherent Subsystem Cache Coherent Interconnect DIRU 0 DIRU 1 CMIU D0 CMIU D1 OBS Video NoC HDMI VidEnc VidDec GP2D Modem DMA PSI PCIe Crypto MCU Audio USB3 USB2 Debug WiFi BT SBM IRQ Main Interconnect Sensor UFS Other IO Sys ctrl Flash eMMC SATA NFC NVM USB3 SatNav CPU big dbg CPU little dbg ISP DSI CSI CSR NoC ROB Memory NoC Sched 0 Sched 1 Sched 2 Sched 3 SRAM DRAM • ArChip16 Example: Large SoCs have multiple classes of interconnect – Non-coherent, Coherent, Control/Status, Observability, etc. • Ncore & FlexNoC interconnects are managed separately from IP blocks, increasing design flexibility ARTERIS IP ISPD 2018, 28 March 2018 Copyright © 2018 Arteris IP | 9 *See* Physical Interconnect Aware Network Optimizer, http://www.ispd.cc/slides/2018/s7_2.pdf, at slide 9.

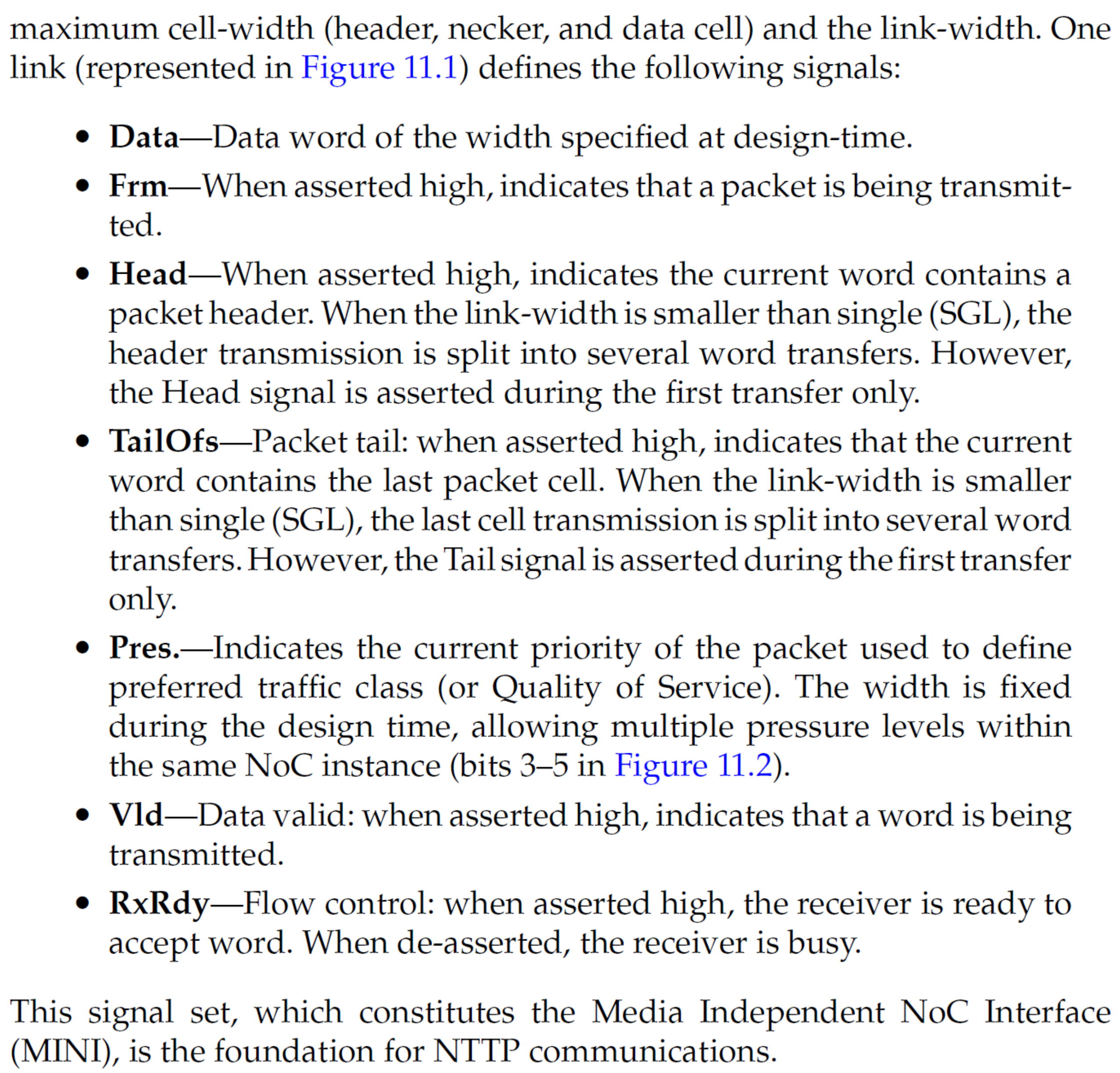
'449 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	The Snapdragon SoC included in the Motorola product utilizes the Arteris NoC to exchange messages over connections via a network, wherein said connections comprises a set of communication channels that are independently configurable. For example, in the the Arteris NoC, "[a]n NTTP transaction is typically made of request packets, traveling through the request network between the master and the slave NIUs, and response packets that are exchanged between a slave NIU and a master NIU through the response network…. Transactions are handed off to the transport layer, which is responsible for delivering packets between endpoints of the NoC (using links, routers, muxes, rated adapters, FIFOs, etc.). Between NoC components, packets are physically transported as cells across various interfaces, a cell being a basic data unit being transported. This is illustrated in Figure 11.1, with one master and one slave node, and one router in the request and response path."

'449 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 Transaction Transport Physical Physical Initiator NIU Rx Tx Link Request packets Response packets Tx Rx Target NIU Request network Response network Tx Data Frm Head TallOfs Pres. Vid RxRdy Rx **FIGURE 11.1** NTTP protocol layers mapped on NoC units and Media Independent NoC Interface—MINI. *See* Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 312-313. Connections within the Arteris NoC network may be defined by a connectivity table:

U.S. Patent No. 7,373,449 (Radulescu and Goossens)
“Apparatus and method for communicating in an integrated circuit”

’449 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 Connectivity Map → Interconnect Connections → Layout DDR_PHY_3 SatNav DDR_PHY_2 MCU Sched3 Sched2 Crypto Sensor USB3 VidDec Audio Modem CPULittle CPUBig USB2 VidEnc PCIe HDMI GPU ISP ZRAM DSI GP2D NVM SRAM SATA DMA Sched1 Sched0 DSP DDR_PHY_1 DDR_PHY_0 Column Clr Row Modifications Defer BTTarg/T/0 CPUBigDebug/T/0 CPULittleDebug/T/0 EMMC/T/0 Flash/T/0 NFC/T/0 NVM/T/0 OtherIO/T/0 PCIeTarg/T/0 PSITarg/T/0 SATA/T/0 SatNav/T/0 Sensor/T/0 SysCtrl/T/0 To CSRNoC/T/0 To CohNo CCSR/T/0 To CohNo CData/T/0 To MemNoC/T/0 USB3Targ/T/0 UniPro/T/0 WiFiTarg/T/0 svc/T/0 Audio/I/0 BTInit/I/0 CPUbigAIU/I/0 CPUlittleAIU/I/0 Crypto/I/0 DMA/I/0 DSPAIU/I/0 Debug/I/0 FromVideoNoC/I/0 GPUAIU/I/0 MCU/I/0 Modem/I/0 PCIeInit/I/0 PSIInit/I/0 USB2/I/0 USB3Init/I/0 WiFiInit/I/0 DC-Topographical • Connectivity table defines interconnect connections within the floorplan • Routes must pass through available channels in the floorplan • Connectivity passes from initiator NIU to switch, to link, to RC buffers and finally to target NIU ARTERIS IP ISPD 2018, 28 March 2018 Copyright © 2018 Arteris IP | 12 *See* Physical Interconnect Aware Network Optimizer, http://www.ispd.cc/slides/2018/s7_2.pdf, at slide 12. In the Arteris NoC, “[t]he delivery of packets within the NoC is the responsibility of the physical layer [where the] link size, or width (i.e., number of wires), is set by the designer at design time[and] [o]ne link (represented in Figure 11.1) defines the following signals… Pres.—Indicates the current priority of the packet used to define preferred traffic class (or Quality of Service). The width is fixed during the design time, allowing multiple pressure levels within the same NoC instance (bits 3–5 in Figure 11.2) [and] RxRdy—flow control”:

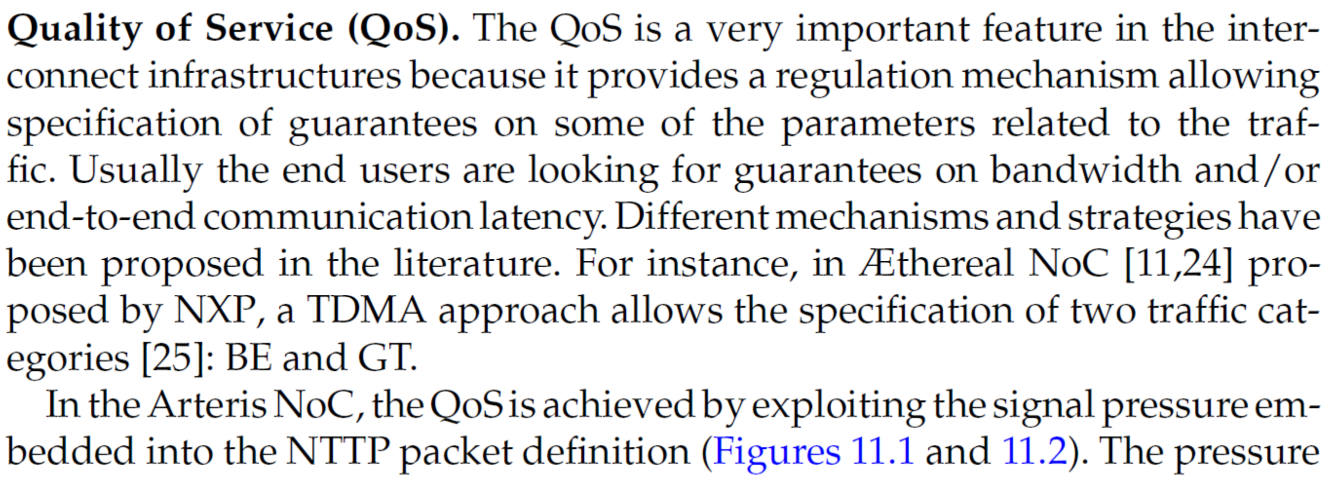
'449 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 ***11.3.1.3 Physical Layer*** The delivery of packets within the NoC is the responsibility of the physical layer. Packets, which have been split by the transport layer into cells, are delivered as words that are sent along links. Within a single clock cycle, the physical layer may carry words comprising a fraction of a cell, a single cell, or multiple cells. The link size, or width (i.e., number of wires), is set by the designer at design time and determines the number of cells of one word. NTTP defines five possible link-widths: quarter (QRT), half (HLF), single (SGL), double (DBL), and quad (QUAD). A single-width (SGL) link transmits one cell per clock cycle, a double-width link transmits two cells per clock cycle, and so on. Words travel within point-to-point links, which are independent from other protocol layers: a word is sent through a transmit port, Tx, over a link to a receive port, Rx. The actual number of wires in a link depends on the

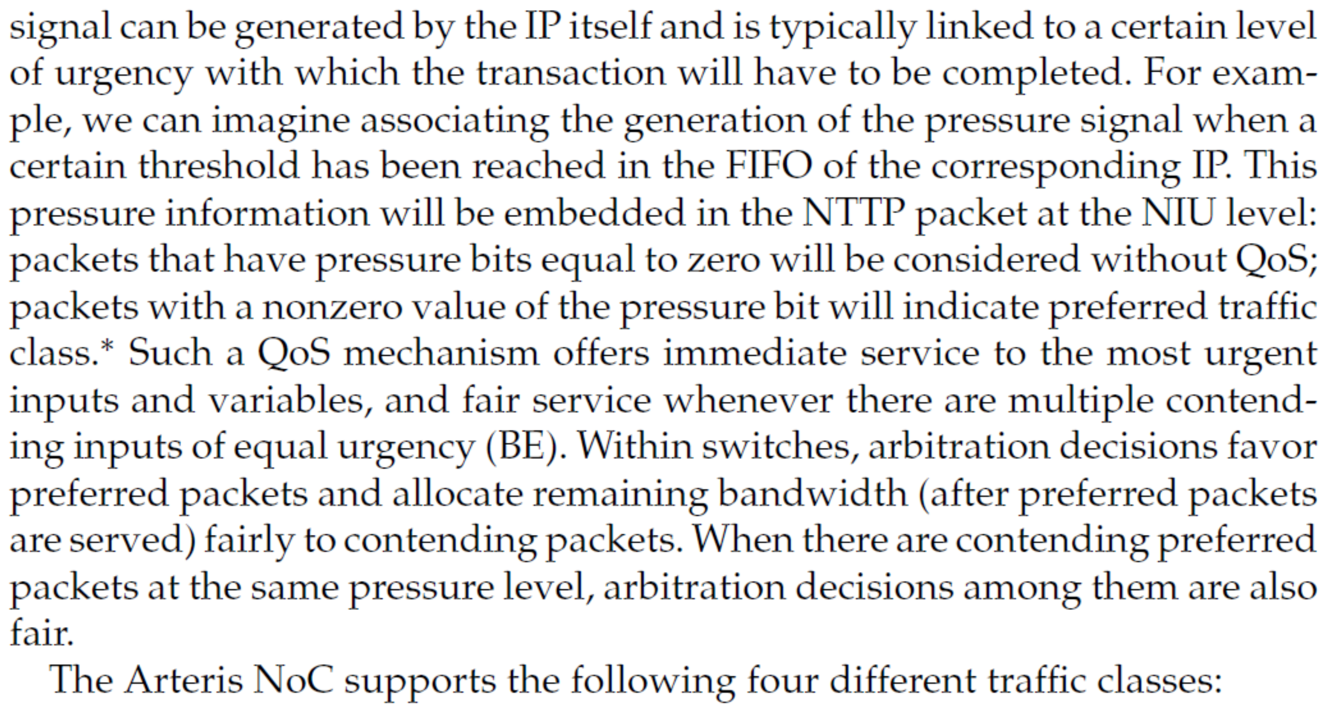


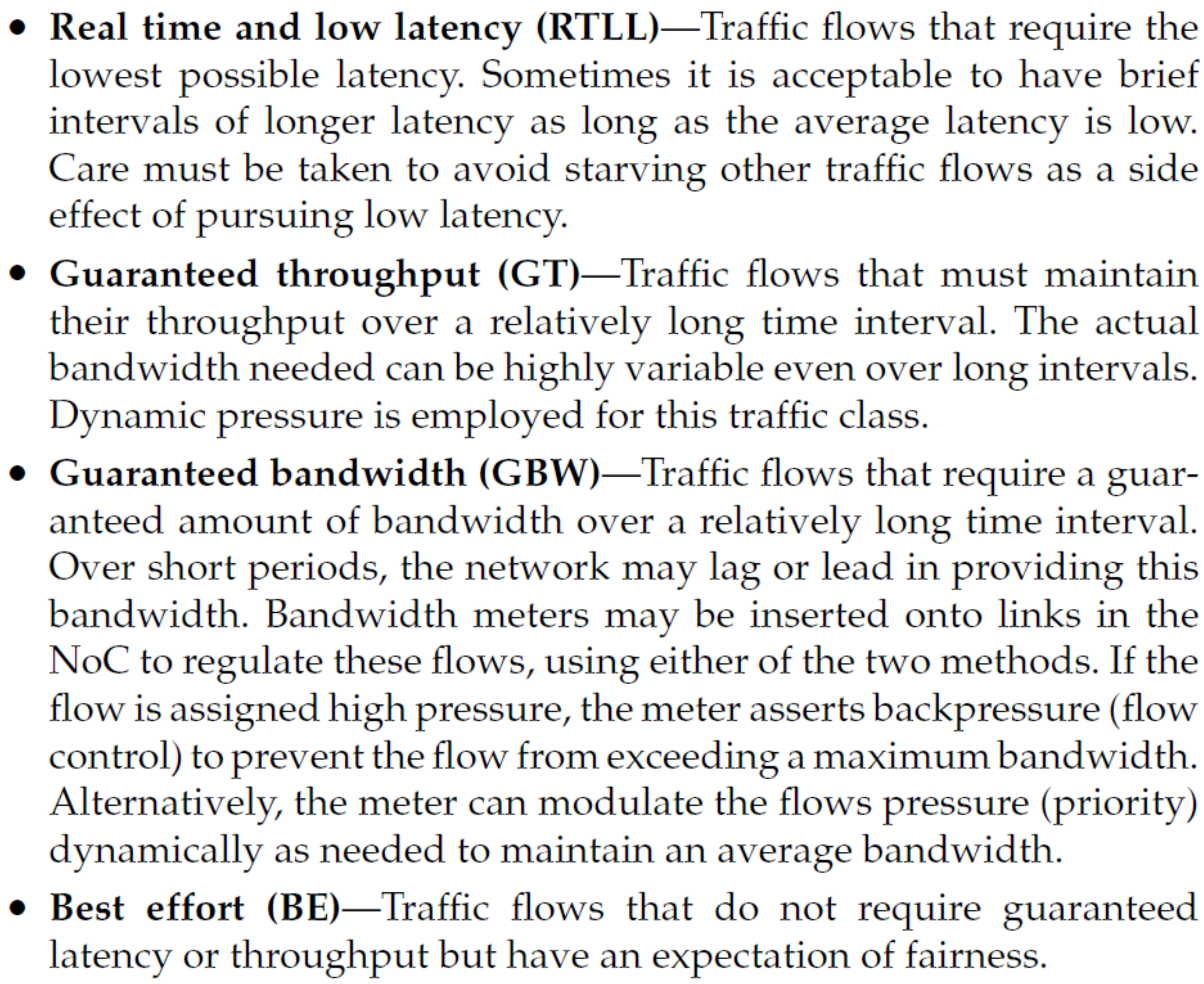
maximum cell-width (header, necker, and data cell) and the link-width. One link (represented in Figure 11.1) defines the following signals:

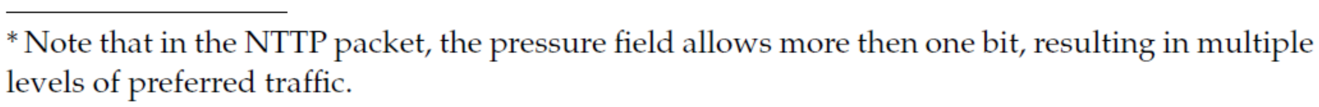
- **Data**—Data word of the width specified at design-time.
- **Frm**—When asserted high, indicates that a packet is being transmitted.
- **Head**—When asserted high, indicates the current word contains a packet header. When the link-width is smaller than single (SGL), the header transmission is split into several word transfers. However, the Head signal is asserted during the first transfer only.
- **TailOfs**—Packet tail: when asserted high, indicates that the current word contains the last packet cell. When the link-width is smaller than single (SGL), the last cell transmission is split into several word transfers. However, the Tail signal is asserted during the first transfer only.
- **Pres.**—Indicates the current priority of the packet used to define preferred traffic class (or Quality of Service). The width is fixed during the design time, allowing multiple pressure levels within the same NoC instance (bits 3–5 in Figure 11.2).
- **Vld**—Data valid: when asserted high, indicates that a word is being transmitted.
- **RxRdy**—Flow control: when asserted high, the receiver is ready to accept word. When de-asserted, the receiver is busy.

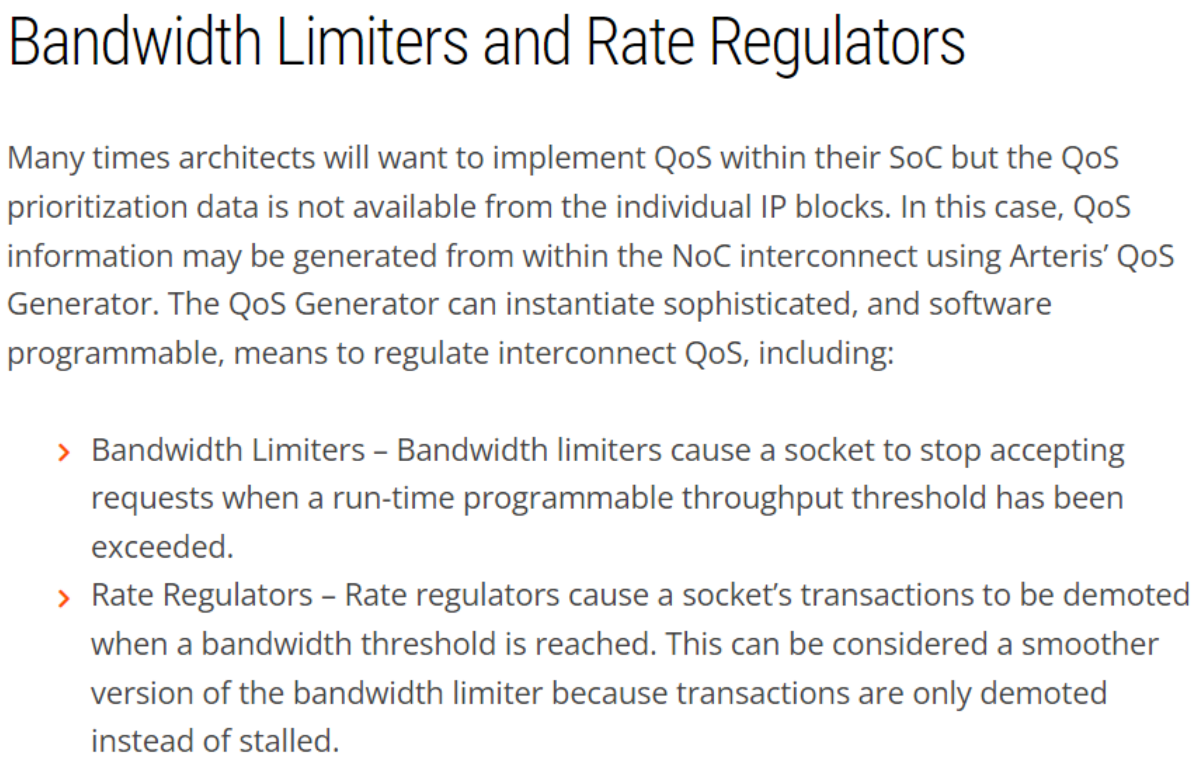
This signal set, which constitutes the Media Independent NoC Interface (MINI), is the foundation for NTTP communications.

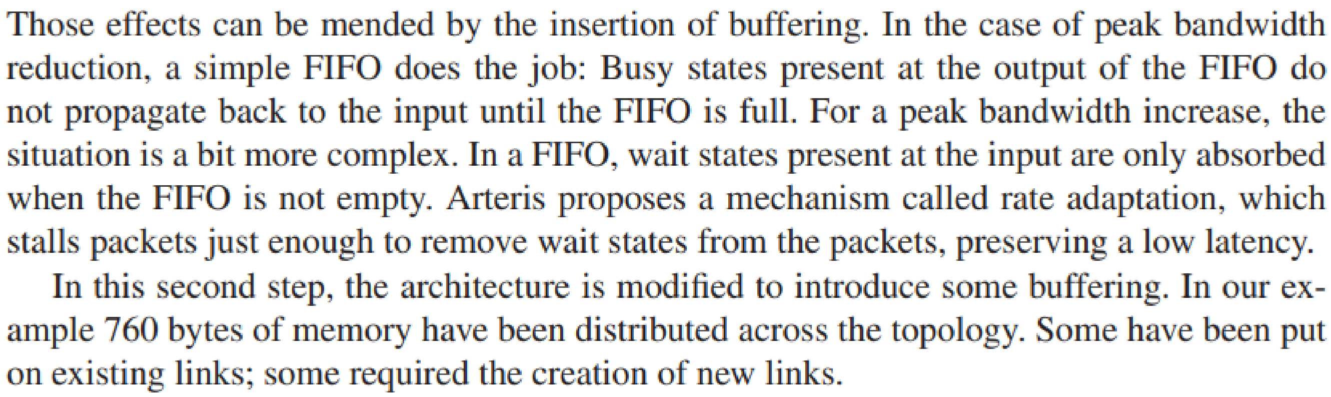
’449 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	See Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 313-314. The Snapdragon SoC included in the Motorola product utilizes the Arteris NoC’s connections that comprise a set of communication channels each having a set of connection properties, any communication channel being independently configurable. For example, as noted above, in the Arteris NoC, “[o]ne link (represented in Figure 11.1) defines the following signals… Pres.—Indicates the current priority of the packet used to define preferred traffic class (or Quality of Service) [and] RxRdy—flow control.” In the Arteris NoC implements Quality of Service (QoS) to “provides a regulation mechanism allowing specification of guarantees on some of the parameters related to the traffic”:  **Quality of Service (QoS).** The QoS is a very important feature in the interconnect infrastructures because it provides a regulation mechanism allowing specification of guarantees on some of the parameters related to the traffic. Usually the end users are looking for guarantees on bandwidth and/or end-to-end communication latency. Different mechanisms and strategies have been proposed in the literature. For instance, in Æthereal NoC [11,24] proposed by NXP, a TDMA approach allows the specification of two traffic categories [25]: BE and GT. In the Arteris NoC, the QoS is achieved by exploiting the signal pressure embedded into the NTTP packet definition (Figures 11.1 and 11.2). The pressure

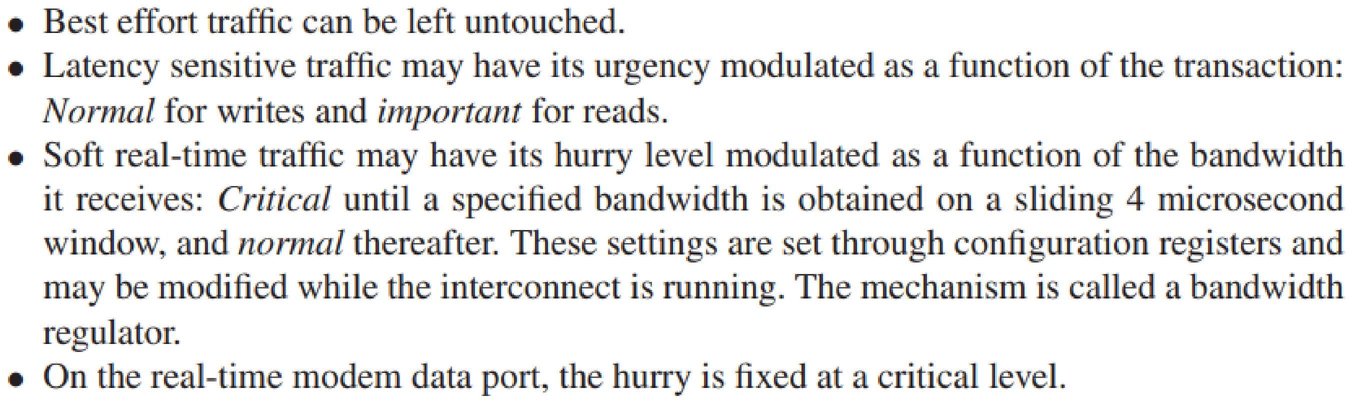
'449 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 signal can be generated by the IP itself and is typically linked to a certain level of urgency with which the transaction will have to be completed. For example, we can imagine associating the generation of the pressure signal when a certain threshold has been reached in the FIFO of the corresponding IP. This pressure information will be embedded in the NTTP packet at the NIU level: packets that have pressure bits equal to zero will be considered without QoS; packets with a nonzero value of the pressure bit will indicate preferred traffic class.* Such a QoS mechanism offers immediate service to the most urgent inputs and variables, and fair service whenever there are multiple contending inputs of equal urgency (BE). Within switches, arbitration decisions favor preferred packets and allocate remaining bandwidth (after preferred packets are served) fairly to contending packets. When there are contending preferred packets at the same pressure level, arbitration decisions among them are also fair. The Arteris NoC supports the following four different traffic classes:

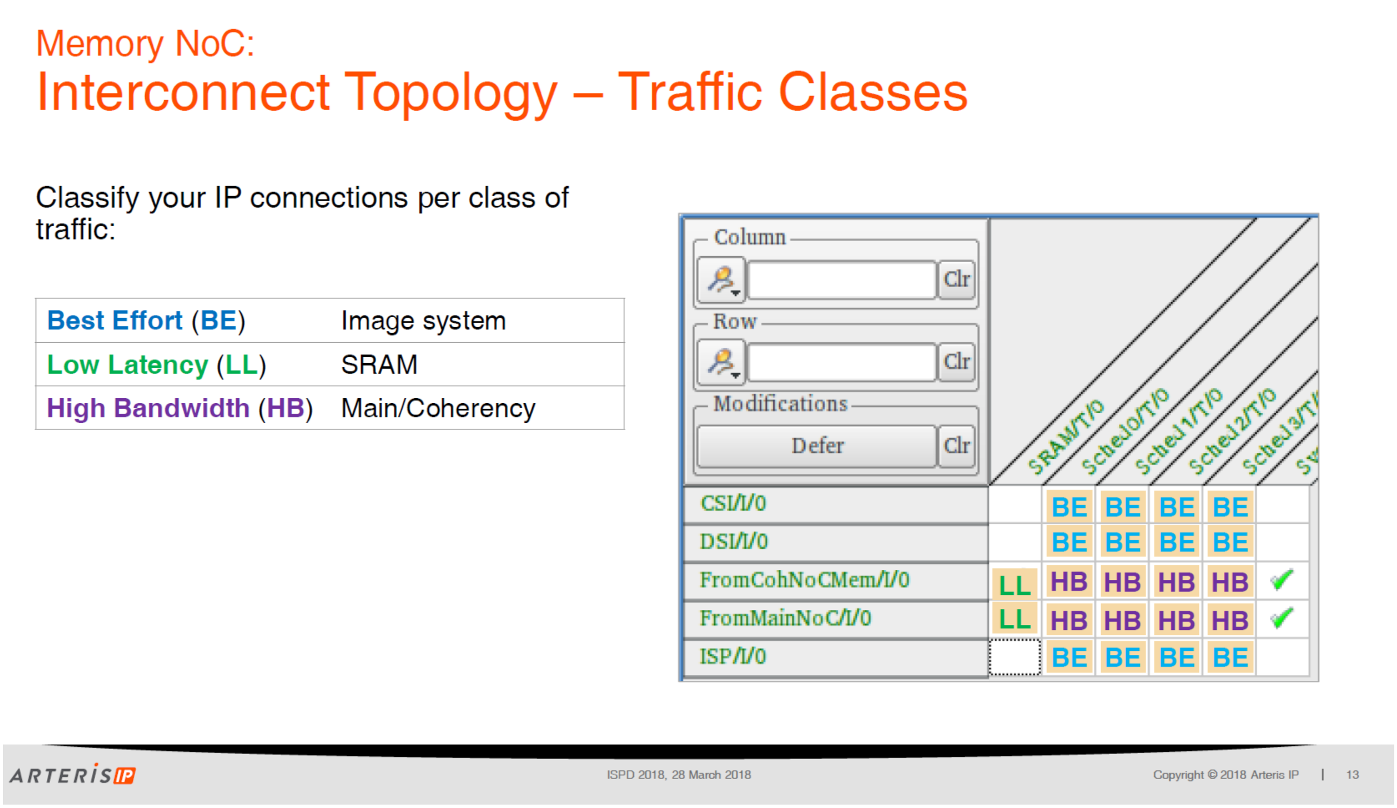
'449 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 • **Real time and low latency (RTLL)**—Traffic flows that require the lowest possible latency. Sometimes it is acceptable to have brief intervals of longer latency as long as the average latency is low. Care must be taken to avoid starving other traffic flows as a side effect of pursuing low latency. • **Guaranteed throughput (GT)**—Traffic flows that must maintain their throughput over a relatively long time interval. The actual bandwidth needed can be highly variable even over long intervals. Dynamic pressure is employed for this traffic class. • **Guaranteed bandwidth (GBW)**—Traffic flows that require a guaranteed amount of bandwidth over a relatively long time interval. Over short periods, the network may lag or lead in providing this bandwidth. Bandwidth meters may be inserted onto links in the NoC to regulate these flows, using either of the two methods. If the flow is assigned high pressure, the meter asserts backpressure (flow control) to prevent the flow from exceeding a maximum bandwidth. Alternatively, the meter can modulate the flows pressure (priority) dynamically as needed to maintain an average bandwidth. • **Best effort (BE)**—Traffic flows that do not require guaranteed latency or throughput but have an expectation of fairness.

'449 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 * Note that in the NTTP packet, the pressure field allows more then one bit, resulting in multiple levels of preferred traffic. Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 315-316. As a further illustration, the Arteris NoC "addresses … varied QoS needs in many ways," including "Dynamic Packet Priorities" and "Dynamic Pressure Propagation": Arbitration: Dynamic Packet Priorities & Dynamic Pressure Propagation Arteris Network on Chip technology addresses these varied QoS needs in many ways: First, the interconnect assigns priorities to transactions to ensure they arrive at the target in the proper order to meet system requirements. Priority levels can be attached to individual packets or to all transactions pending on a socket. The interconnect can also assign Dynamic Packet Priorities at runtime. Second, the interconnect can sense when high priority packets may be blocked or slowed due to downstream traffic congestion and can then clear a path for these high priority packets. This technology, called Dynamic Pressure Propagation, is analogous to a fire truck racing down city streets: All traffic pulls to the side of the road to let the fire truck through. https://www.arteris.com/end-to-end-quality-of-service-qos

'449 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	As a further illustration, "QoS information may be generated from within the [Arteris] NoC interconnect using Arteris' QoS Generator":  Bandwidth Limiters and Rate Regulators Many times architects will want to implement QoS within their SoC but the QoS prioritization data is not available from the individual IP blocks. In this case, QoS information may be generated from within the NoC interconnect using Arteris' QoS Generator. The QoS Generator can instantiate sophisticated, and software programmable, means to regulate interconnect QoS, including: › Bandwidth Limiters – Bandwidth limiters cause a socket to stop accepting requests when a run-time programmable throughput threshold has been exceeded. › Rate Regulators – Rate regulators cause a socket's transactions to be demoted when a bandwidth threshold is reached. This can be considered a smoother version of the bandwidth limiter because transactions are only demoted instead of stalled. https://www.arteris.com/end-to-end-quality-of-service-qos As a further illustration, the Arteris NoC uses "a mechanism called rated adaptation, which stalls packets just enough to remove wait states from the packets, preserving a low latency." For other traffic, the "[b]est effort traffic can be left untouched[,]" "[l]atency sensitive traffic may have its

'449 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	urgency modulated as a function of the transaction[,]" "[s]oft real-time traffic may have its hurry level modulated as a function of the bandwidth it receives[,]" and "[o]n the real-time modem data port, the hurry is fixed at a critical level":  Those effects can be mended by the insertion of buffering. In the case of peak bandwidth reduction, a simple FIFO does the job: Busy states present at the output of the FIFO do not propagate back to the input until the FIFO is full. For a peak bandwidth increase, the situation is a bit more complex. In a FIFO, wait states present at the input are only absorbed when the FIFO is not empty. Arteris proposes a mechanism called rate adaptation, which stalls packets just enough to remove wait states from the packets, preserving a low latency. In this second step, the architecture is modified to introduce some buffering. In our example 760 bytes of memory have been distributed across the topology. Some have been put on existing links; some required the creation of new links. See Application driven network-on-chip architecture exploration & refinement for a complex SoC, https://www.arteris.com/hs-fs/hub/48858/file-14363521-pdf/docs/springerappdrivennocarchitecture8.5x11.pdf, at pg.16. For the other traffic, "the configuration can be done in architecture":

'449 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 • Best effort traffic can be left untouched. • Latency sensitive traffic may have its urgency modulated as a function of the transaction: *Normal* for writes and *important* for reads. • Soft real-time traffic may have its hurry level modulated as a function of the bandwidth it receives: *Critical* until a specified bandwidth is obtained on a sliding 4 microsecond window, and *normal* thereafter. These settings are set through configuration registers and may be modified while the interconnect is running. The mechanism is called a bandwidth regulator. • On the real-time modem data port, the hurry is fixed at a critical level. *Id.* at 18. As a further illustration, connections within the Arteris NoC may be classified by traffic class and traffic classes may be mapped onto the Arteris interconnect topology:

'449 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	

Memory NoC:

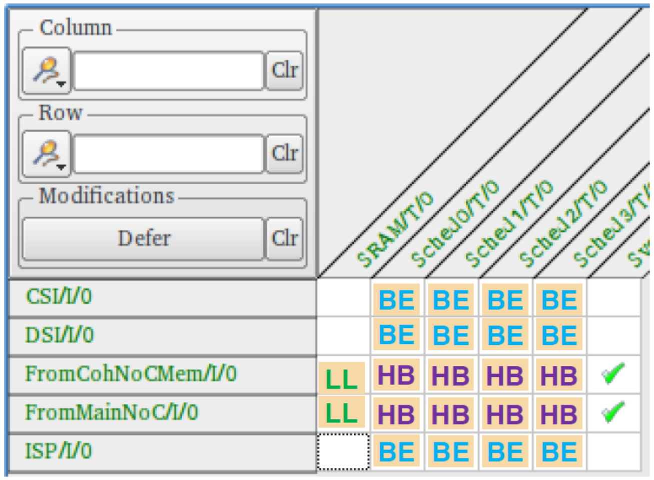
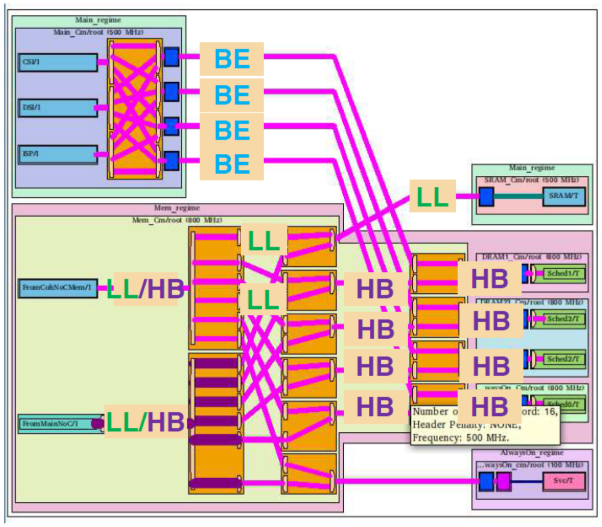
Interconnect Topology – Traffic Classes

Classify your IP connections per class of traffic:

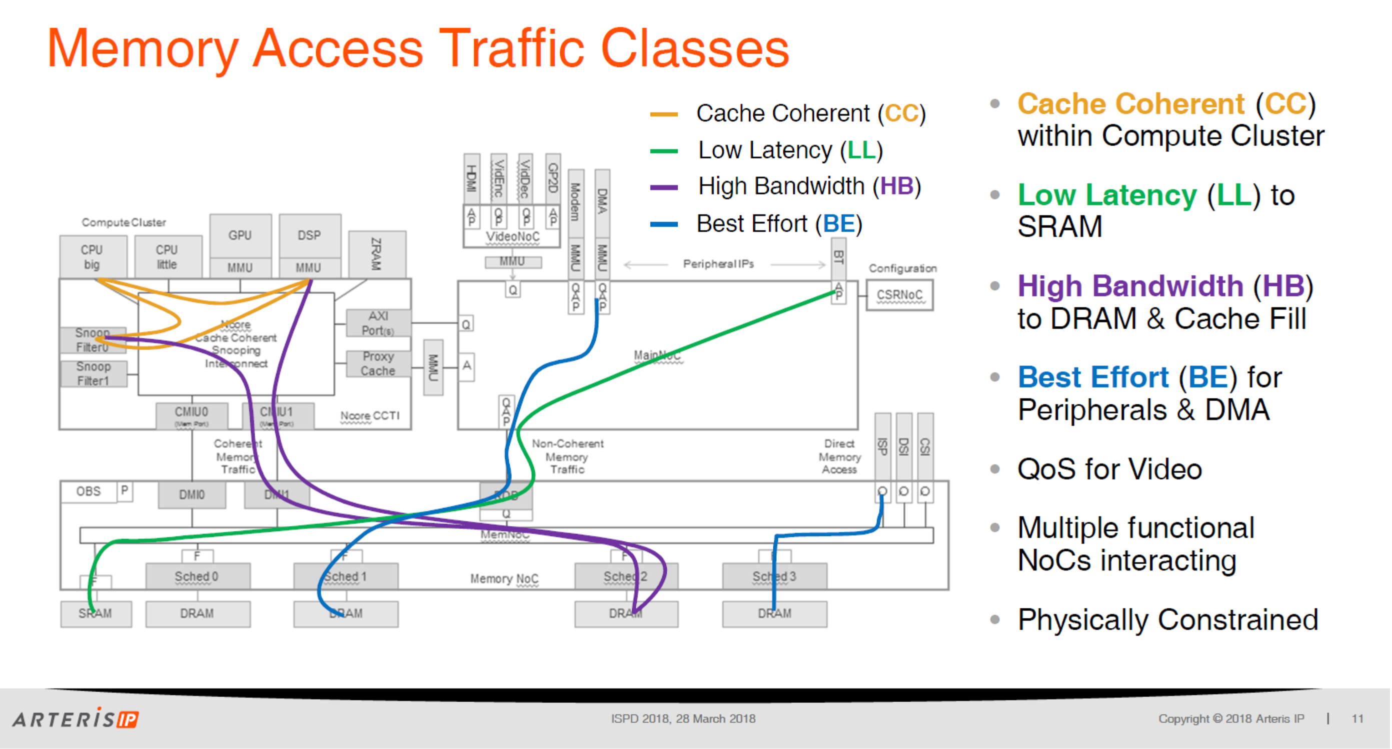
Best Effort (BE)	Image system
Low Latency (LL)	SRAM
High Bandwidth (HB)	Main/Coherency

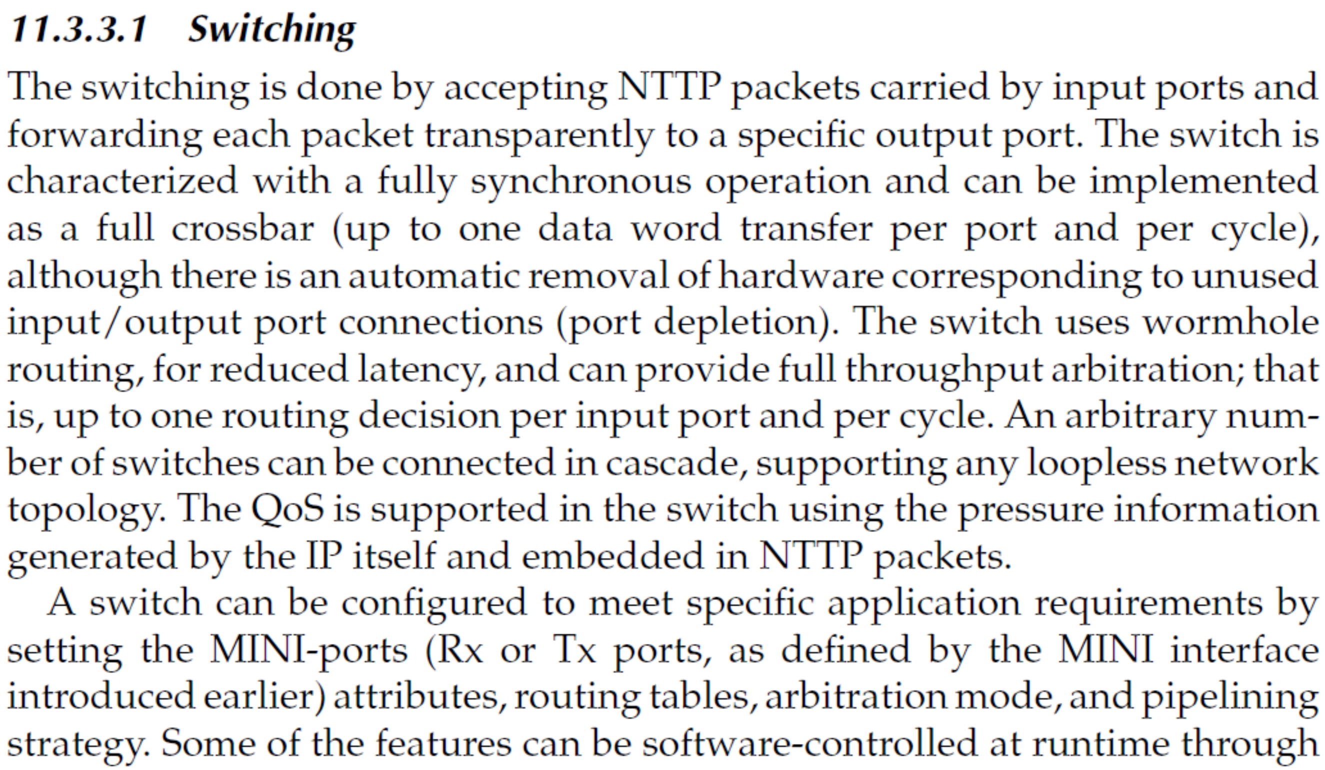
	SRAM/T/0	Sched0/T/0	Sched1/T/0	Sched2/T/0	Sched3/T/	Sy
CSI/I/0		BE	BE	BE	BE	
DSI/I/0		BE	BE	BE	BE	
FromCohNoCMem/I/0	LL	HB	HB	HB	HB	✓
FromMainNoC/I/0	LL	HB	HB	HB	HB	✓
ISP/I/0		BE	BE	BE	BE	

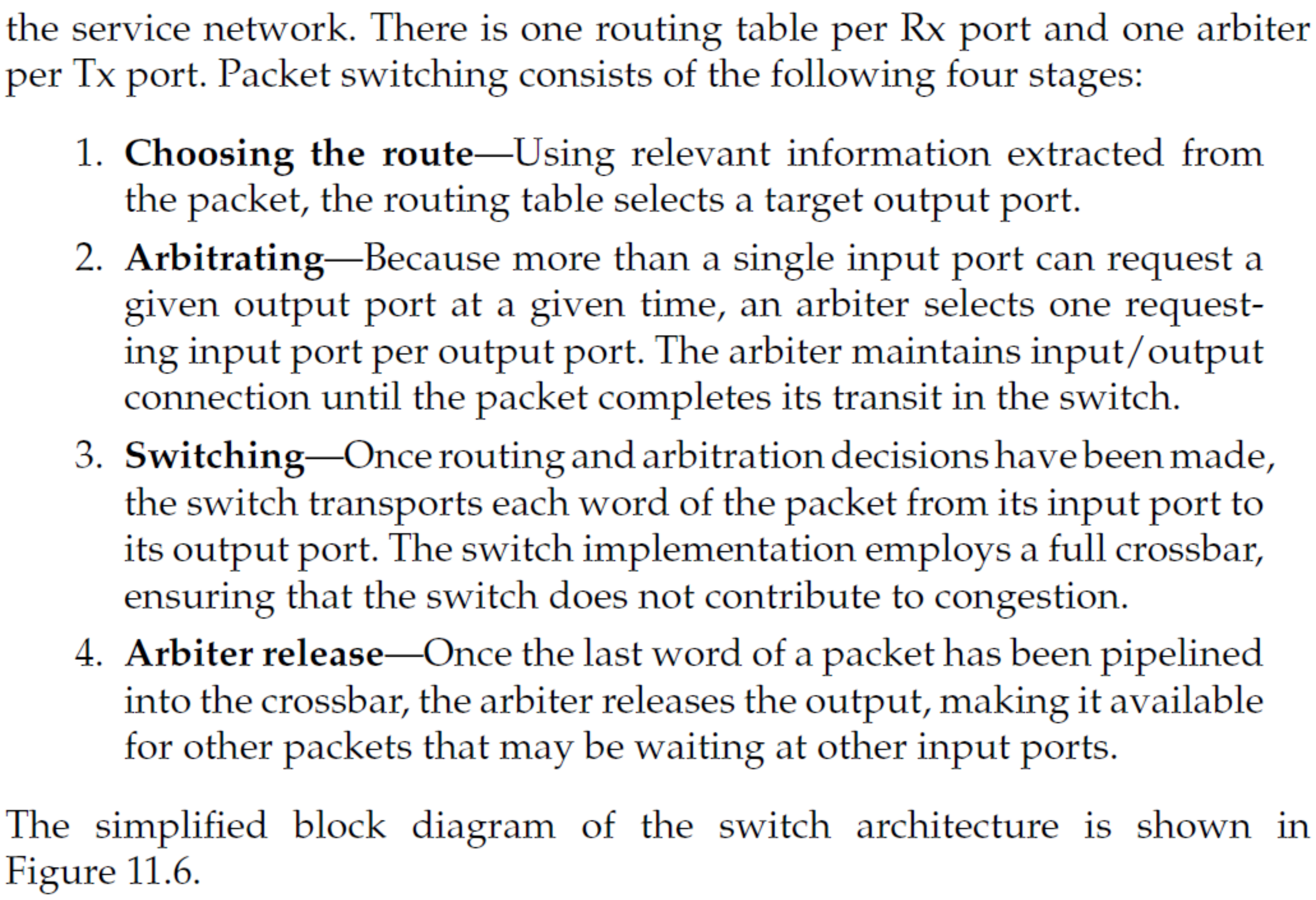
U.S. Patent No. 7,373,449 (Radulescu and Goossens)
"Apparatus and method for communicating in an integrated circuit"

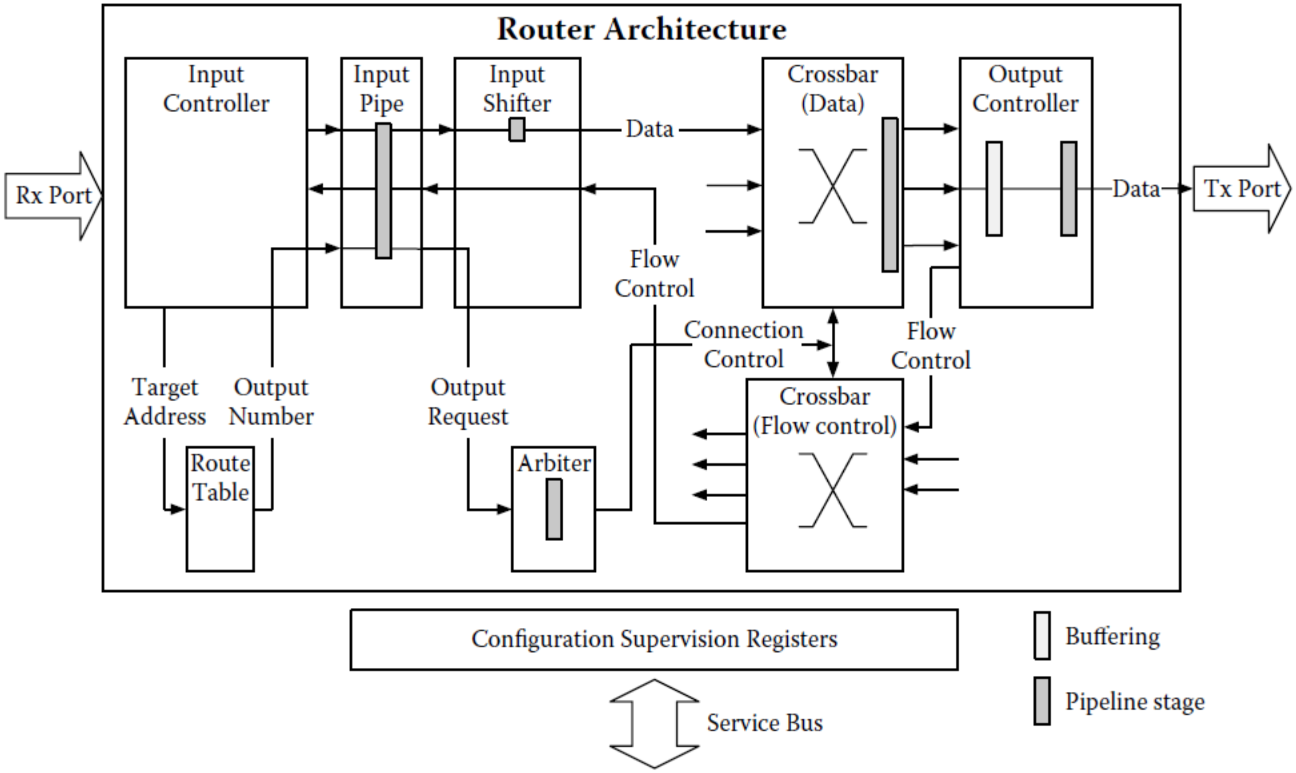
<table>
<tr><th>'449 Patent Claim</th><th>Motorola Product Including Snapdragon System on Chip[1]</th></tr>
<tr><td></td><td>
Memory NoC:
Traffic classes are mapped onto logical interconnect topology

<table>
<tr><th>Column / Row / Modifications: Defer</th><th>SRAM/T/0</th><th>Sched0/T/0</th><th>Sched1/T/0</th><th>Sched2/T/0</th><th>Sched3/T/</th><th>Sv</th></tr>
<tr><td>CSI/I/0</td><td></td><td>BE</td><td>BE</td><td>BE</td><td>BE</td><td></td></tr>
<tr><td>DSI/I/0</td><td></td><td>BE</td><td>BE</td><td>BE</td><td>BE</td><td></td></tr>
<tr><td>FromCohNoCMem/I/0</td><td>LL</td><td>HB</td><td>HB</td><td>HB</td><td>HB</td><td>✓</td></tr>
<tr><td>FromMainNoC/I/0</td><td>LL</td><td>HB</td><td>HB</td><td>HB</td><td>HB</td><td>✓</td></tr>
<tr><td>ISP/I/0</td><td></td><td>BE</td><td>BE</td><td>BE</td><td>BE</td><td></td></tr>
</table>
Column
Clr
Row
Clr
Modifications
Defer
Clr

Main_regime
Main_Cm/root (500 MHz)
CSI/I
DSI/I
ISP/I
BE
BE
BE
BE
Main_regime
SRAM_Cm/root (500 MHz)
LL
SRAM/T
Mem_regime
Mem_Cm/root (800 MHz)
FromCohNoCMem/I
LL/HB
LL
LL
FromMainNoC/I
LL/HB
HB
HB
HB
HB
DRAM1_Cm/root (800 MHz)
Sched1/T
HB
Sched2/T
HB
Sched2/T
HB
...waysOn_Cm/root (800 MHz)
Sched0/T
HB
Number o ... ord: 16,
Header Penalty: NONE,
Frequency: 500 MHz.
AlwaysOn_regime
...waysOn_cm/root (100 MHz)
Svc/T
ARTERIS IP
ISPD 2018, 28 March 2018
Copyright © 2018 Arteris IP | 16
</td></tr>
</table>

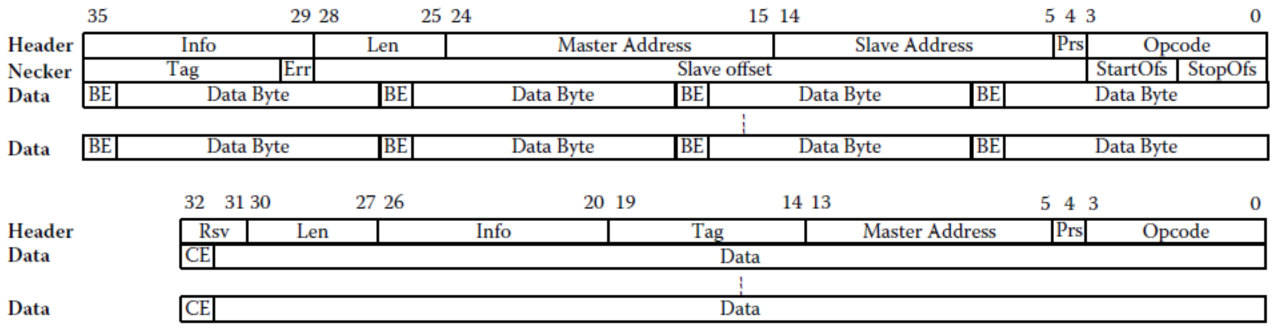
U.S. Patent No. 7,373,449 (Radulescu and Goossens)
"Apparatus and method for communicating in an integrated circuit"

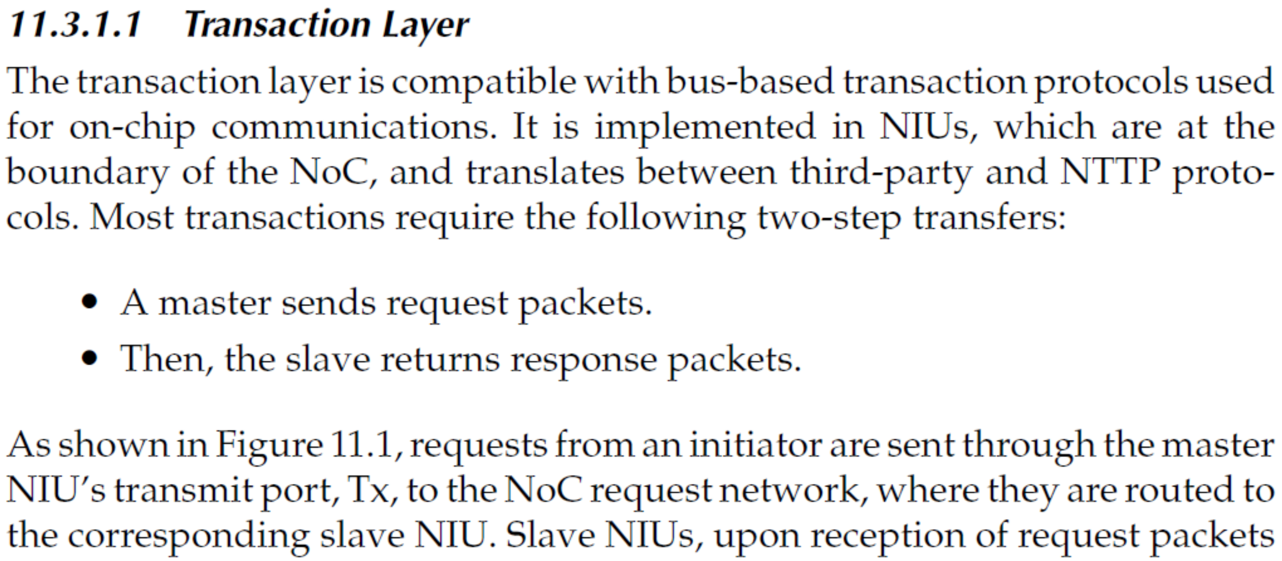
'449 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 Memory Access Traffic Classes Cache Coherent (CC) Low Latency (LL) High Bandwidth (HB) Best Effort (BE) Compute Cluster CPU big CPU little GPU DSP MMU MMU ZRAM Snoop Filter0 Snoop Filter1 Ncore Cache Coherent Snooping Interconnect AXI Port(s) Proxy Cache CMIU0 CMIU1 Ncore CCTI HDMI VidEnc VidDec GP2D AP QP QP AP VideoNoC MMU Modem DMA MMU MMU Peripheral IPs BT Configuration CSRNoC MMU MainNoC Coherent Memory Traffic Non-Coherent Memory Traffic Direct Memory Access ISP DSI CSI OBS P DMI0 DMI1 RDB MemNoC Memory NoC Sched 0 Sched 1 Sched 2 Sched 3 SRAM DRAM DRAM DRAM DRAM - Cache Coherent (CC) within Compute Cluster - Low Latency (LL) to SRAM - High Bandwidth (HB) to DRAM & Cache Fill - Best Effort (BE) for Peripherals & DMA - QoS for Video - Multiple functional NoCs interacting - Physically Constrained ARTERIS IP ISPD 2018, 28 March 2018 Copyright © 2018 Arteris IP | 11 *See* Physical Interconnect Aware Network Optimizer, http://www.ispd.cc/slides/2018/s7_2.pdf, at slides 11, 13, 16. As a further illustration, in the Arteris NoC, "QoS is supported in the switch using pressure information generated by the IP itself and embedded in NTTP packets" and "[s]ome features [of the switch] can be software-controlled at runtime through the service network":

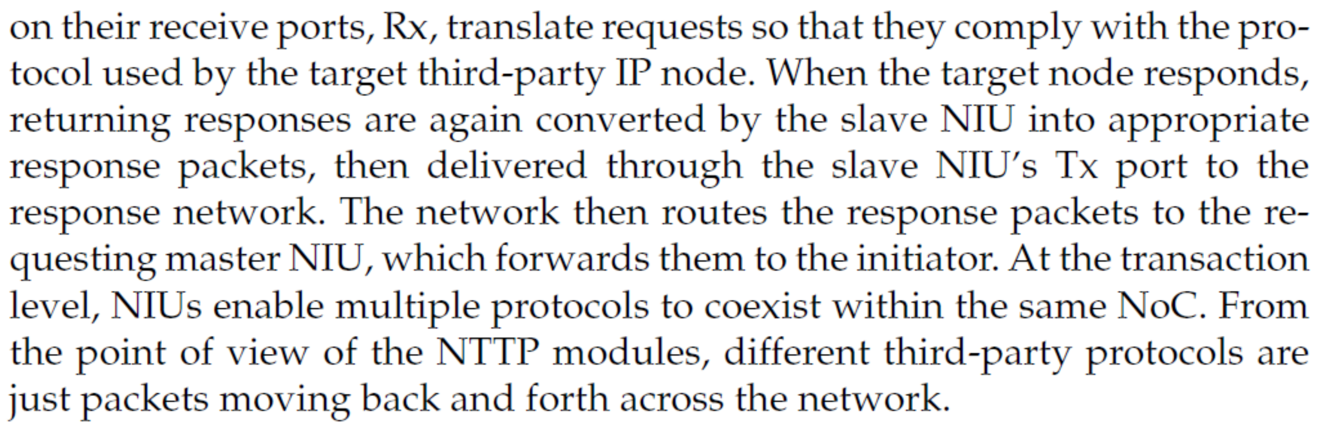
'449 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 ***11.3.3.1 Switching*** The switching is done by accepting NTTP packets carried by input ports and forwarding each packet transparently to a specific output port. The switch is characterized with a fully synchronous operation and can be implemented as a full crossbar (up to one data word transfer per port and per cycle), although there is an automatic removal of hardware corresponding to unused input/output port connections (port depletion). The switch uses wormhole routing, for reduced latency, and can provide full throughput arbitration; that is, up to one routing decision per input port and per cycle. An arbitrary number of switches can be connected in cascade, supporting any loopless network topology. The QoS is supported in the switch using the pressure information generated by the IP itself and embedded in NTTP packets. A switch can be configured to meet specific application requirements by setting the MINI-ports (Rx or Tx ports, as defined by the MINI interface introduced earlier) attributes, routing tables, arbitration mode, and pipelining strategy. Some of the features can be software-controlled at runtime through

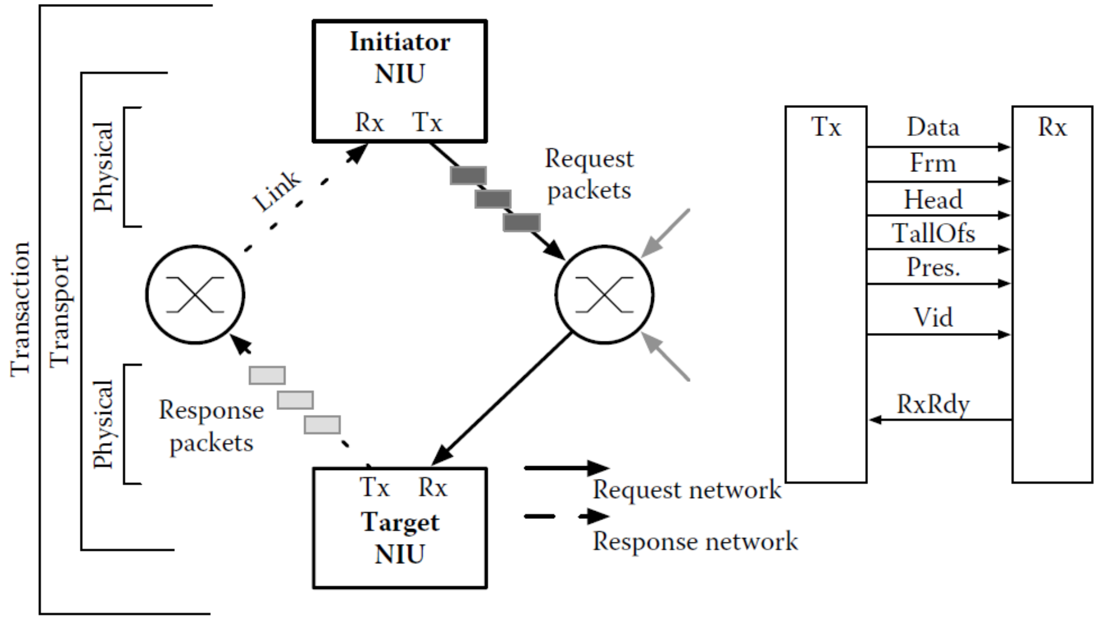
'449 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 the service network. There is one routing table per Rx port and one arbiter per Tx port. Packet switching consists of the following four stages: 1. **Choosing the route**—Using relevant information extracted from the packet, the routing table selects a target output port. 2. **Arbitrating**—Because more than a single input port can request a given output port at a given time, an arbiter selects one requesting input port per output port. The arbiter maintains input/output connection until the packet completes its transit in the switch. 3. **Switching**—Once routing and arbitration decisions have been made, the switch transports each word of the packet from its input port to its output port. The switch implementation employs a full crossbar, ensuring that the switch does not contribute to congestion. 4. **Arbiter release**—Once the last word of a packet has been pipelined into the crossbar, the arbiter releases the output, making it available for other packets that may be waiting at other input ports. The simplified block diagram of the switch architecture is shown in Figure 11.6.

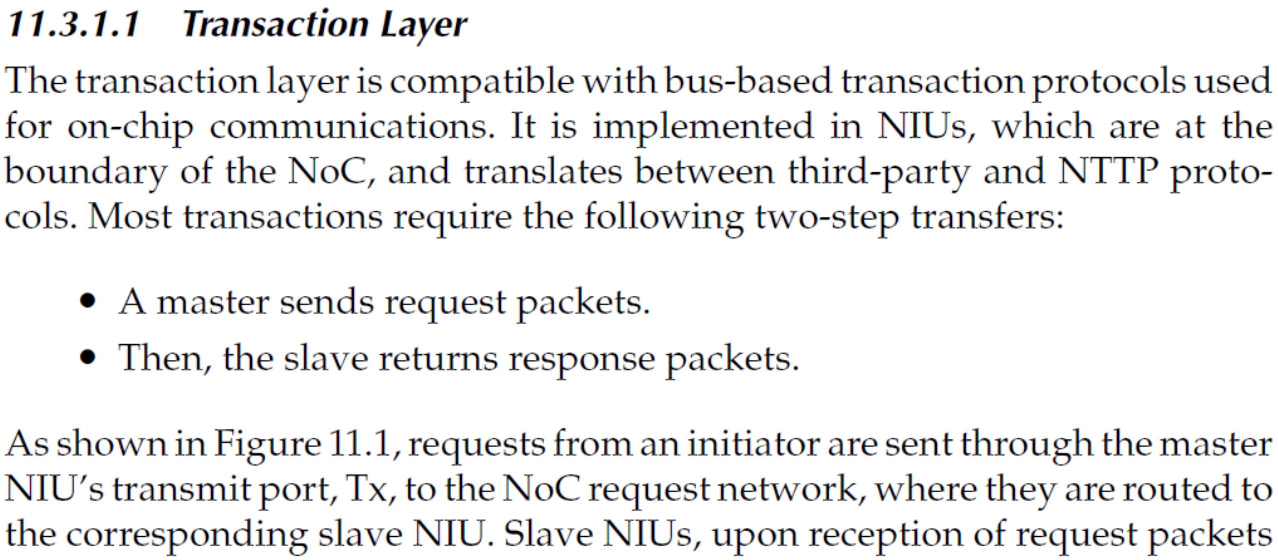
'449 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 Router Architecture Input Controller Input Pipe Input Shifter Data Crossbar (Data) Output Controller Rx Port Data Tx Port Flow Control Connection Control Flow Control Target Address Output Number Output Request Crossbar (Flow control) Route Table Arbiter Configuration Supervision Registers Buffering Pipeline stage Service Bus **FIGURE 11.6** Packet transportation unit: Router architecture. *See* Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 319-320.

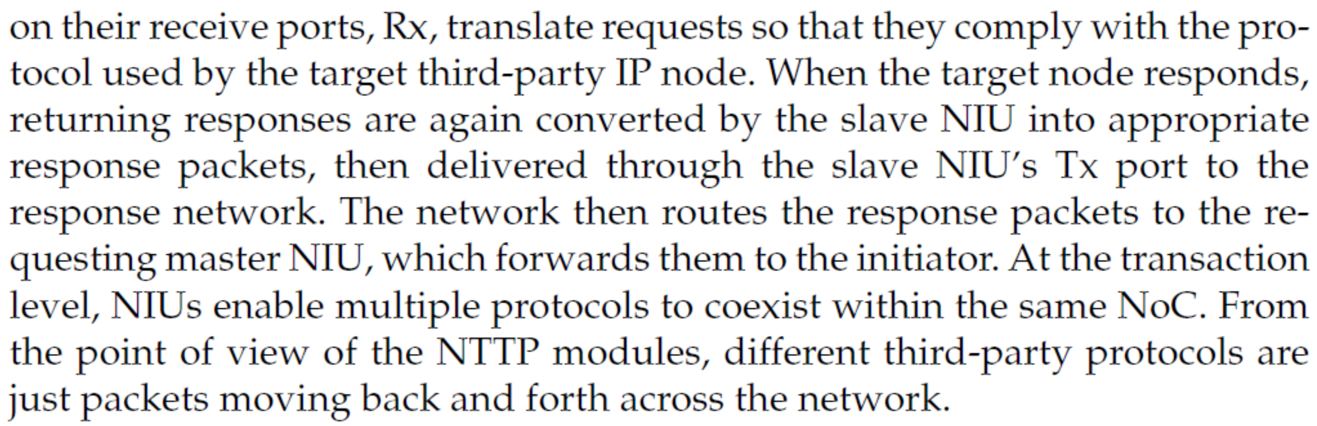
'449 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	As a further illustration, the "Pres." signal in the NTTP packet "[i]ndicates the current priority of the packet used to define preferred traffic class (or Quality of Service). The width is fixed during the design time, allowing multiple pressure levels within the same NoC instance (bits 3–5 in Figure 11.2)."  35 29 28 25 24 15 14 5 4 3 0 Header Info Len Master Address Slave Address Prs Opcode Necker Tag Err Slave offset StartOfs StopOfs Data BE Data Byte 32 31 30 27 26 20 19 14 13 Rsv CE **FIGURE 11.2** NTTP packet structure. *See id.* at 313, 314. As a further illustration, in the Arteris NoC, "the routing tables actually used in the switch are parameterizable for each input port of the switch. It is thus possible to use different routing tables for each switch input. Routing tables can optionally be programmed via the service network interface; in this case, their configuration registers appear in the switch register address map." *See id.* at 322.

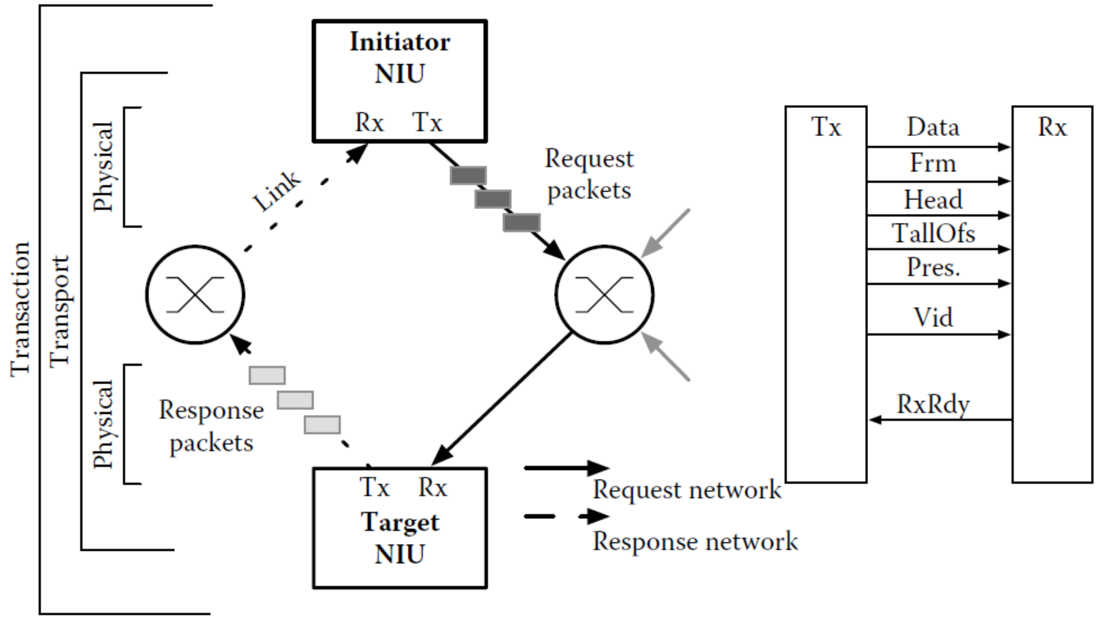
'449 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
wherein said connection through the network supports transactions comprising at least one of outgoing messages from the first module to the second module and return messages from the second module to the first module	Without conceding that the preamble of claim 10 of the '449 Patent is limiting, the Arteris NoC in the Snapdragon SoC included in the Motorola product has connections through the network that support transactions comprising at least one of outgoing messages from the first module to the second module and return messages from the second module to the first module, either literally or under the doctrine of equivalents. For example, in the Arteris NoC used by the Snapdragon SoC included in the Motorola product, "[m]ost transactions require the following two-step transfers," including "[a] master send[ing] request packets" and "the slave return[ing] response packets":  ***11.3.1.1 Transaction Layer*** The transaction layer is compatible with bus-based transaction protocols used for on-chip communications. It is implemented in NIUs, which are at the boundary of the NoC, and translates between third-party and NTTP protocols. Most transactions require the following two-step transfers: - A master sends request packets. - Then, the slave returns response packets. As shown in Figure 11.1, requests from an initiator are sent through the master NIU's transmit port, Tx, to the NoC request network, where they are routed to the corresponding slave NIU. Slave NIUs, upon reception of request packets

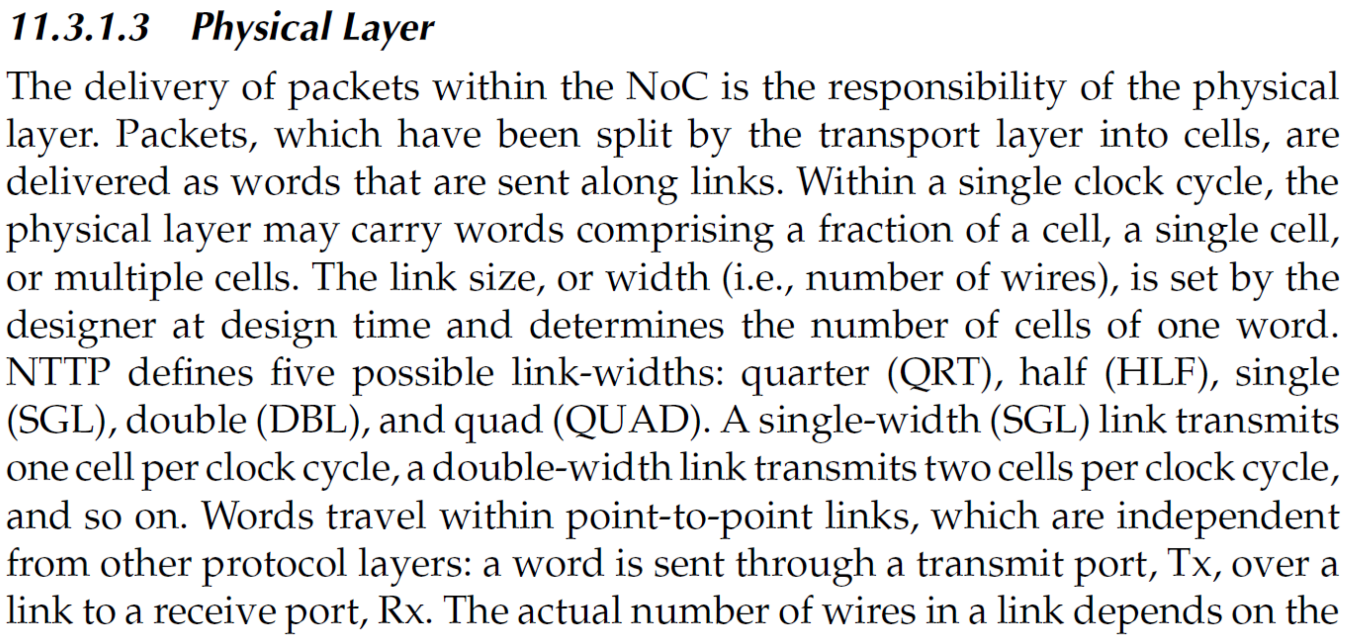
'449 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 on their receive ports, Rx, translate requests so that they comply with the protocol used by the target third-party IP node. When the target node responds, returning responses are again converted by the slave NIU into appropriate response packets, then delivered through the slave NIU's Tx port to the response network. The network then routes the response packets to the requesting master NIU, which forwards them to the initiator. At the transaction level, NIUs enable multiple protocols to coexist within the same NoC. From the point of view of the NTTP modules, different third-party protocols are just packets moving back and forth across the network.

'449 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 Transaction Transport Physical Physical Initiator NIU Rx Tx Link Request packets Response packets Tx Rx Target NIU Request network Response network Tx Rx Data Frm Head TallOfs Pres. Vid RxRdy **FIGURE 11.1** NTTP protocol layers mapped on NoC units and Media Independent NoC Interface—MINI. *See* Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 312-313.
and further comprising the steps of: the first module issuing a request	In the Arteris NoC in the Snapdragon SoC included in the Motorola product, the first module issues a request for a connection with the second module to a communication manager, wherein the request comprises desired connection properties associated with the sets of communication channels, either literally or under the doctrine of equivalents.

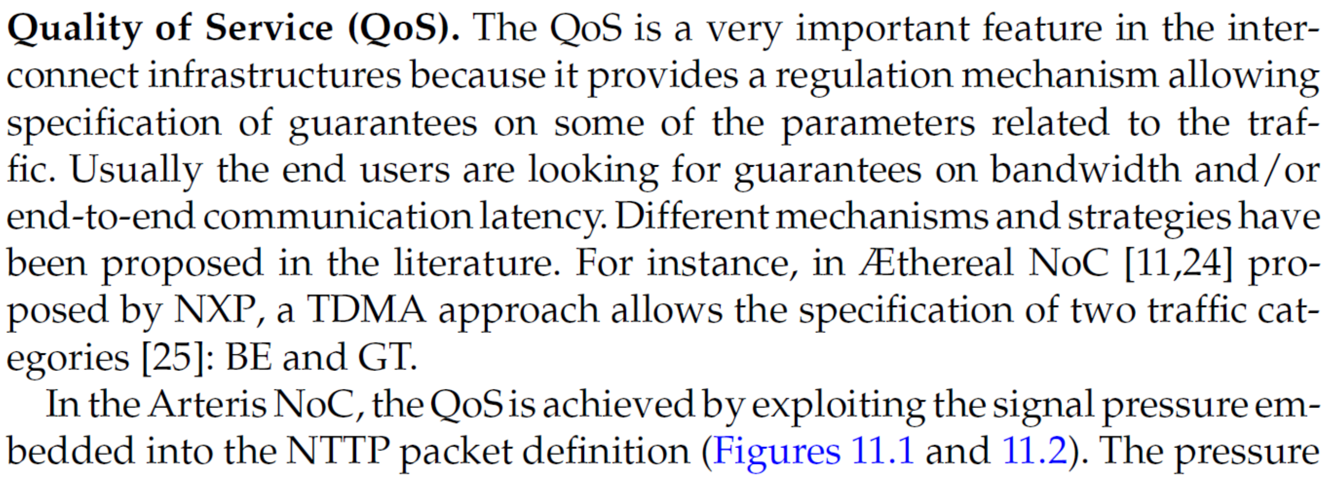
'449 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
for a connection with the second module to a communication manager, wherein the request comprises desired connection properties associated with the sets of communication channels;	The first module of the Snapdragon SoC included in the Motorola product utilizes the Arteris NoC to issue a request for a connection with the second module to a communication manager. For example, in the Arteris NoC, "[m]ost transactions require the following two-step transfers," including "[a] master send[ing] request packets" and "the slave return[ing] response packets":  ***11.3.1.1 Transaction Layer*** The transaction layer is compatible with bus-based transaction protocols used for on-chip communications. It is implemented in NIUs, which are at the boundary of the NoC, and translates between third-party and NTTP protocols. Most transactions require the following two-step transfers: • A master sends request packets. • Then, the slave returns response packets. As shown in Figure 11.1, requests from an initiator are sent through the master NIU's transmit port, Tx, to the NoC request network, where they are routed to the corresponding slave NIU. Slave NIUs, upon reception of request packets

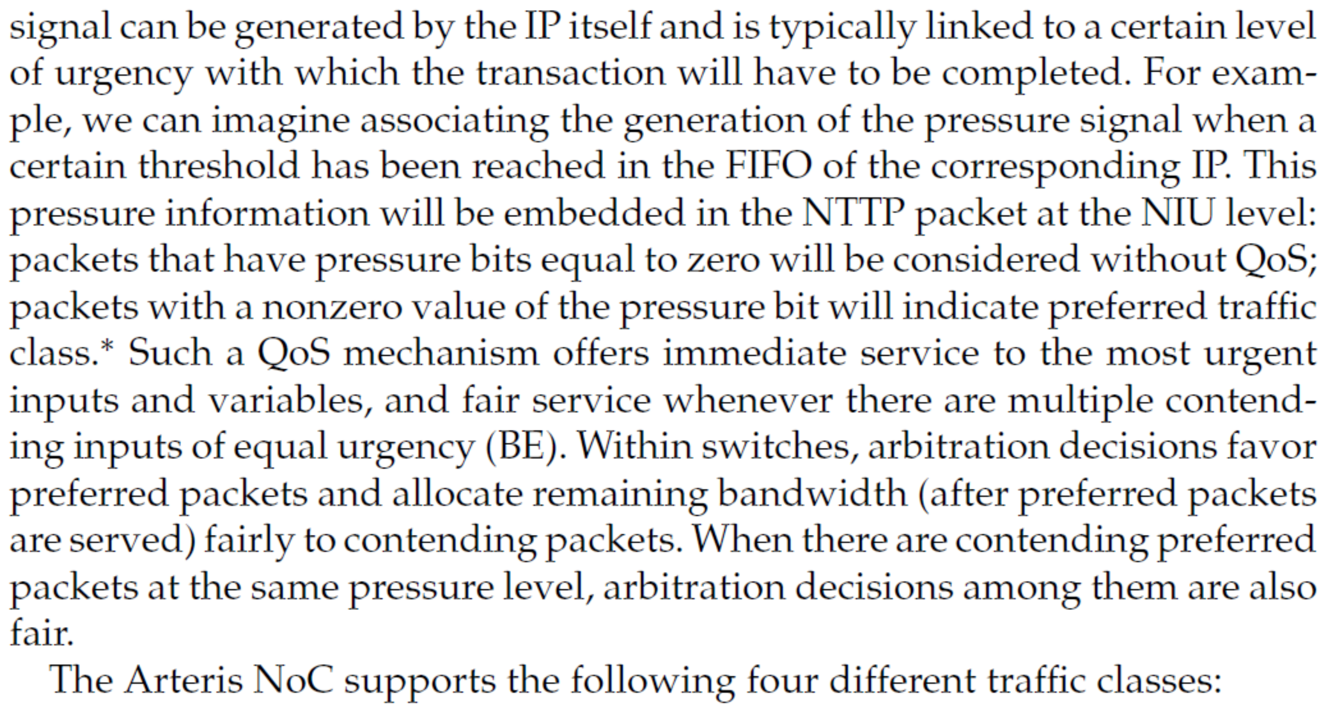
'449 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 on their receive ports, Rx, translate requests so that they comply with the protocol used by the target third-party IP node. When the target node responds, returning responses are again converted by the slave NIU into appropriate response packets, then delivered through the slave NIU's Tx port to the response network. The network then routes the response packets to the requesting master NIU, which forwards them to the initiator. At the transaction level, NIUs enable multiple protocols to coexist within the same NoC. From the point of view of the NTTP modules, different third-party protocols are just packets moving back and forth across the network.

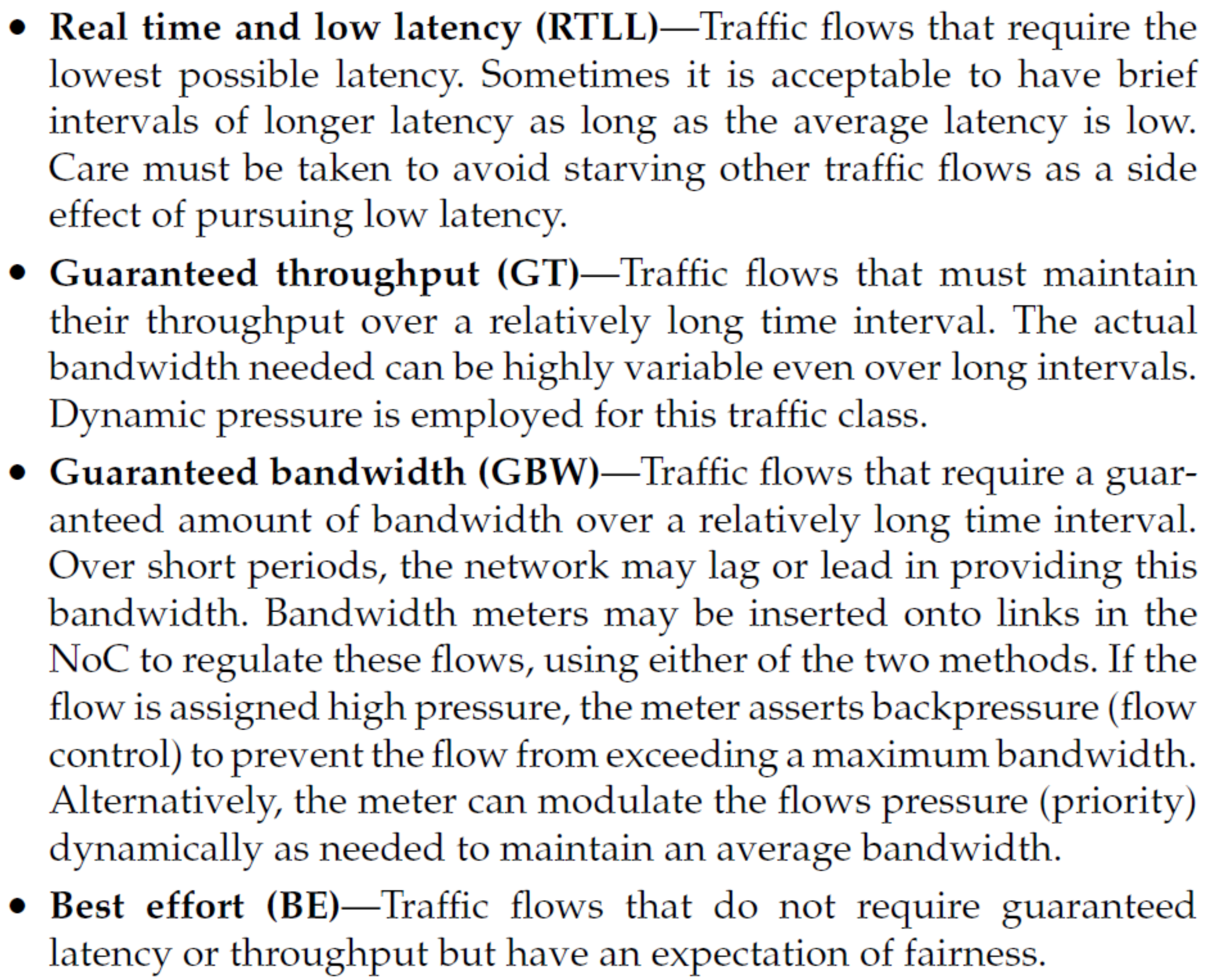
'449 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 Transaction Transport Physical Physical Initiator NIU Rx Tx Link Request packets Response packets Tx Rx Target NIU Request network Response network Tx Data Frm Head TallOfs Pres. Vid RxRdy Rx **FIGURE 11.1** NTTP protocol layers mapped on NoC units and Media Independent NoC Interface—MINI. *See* Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 312-313. The request issued by first module of the Snapdragon SoC included in the Motorola product comprises desired connection properties associated with the sets of communication channels. For example, in the Arteris NoC, "[t]he delivery of packets within the NoC is the responsibility of the physical layer [where the] link size, or width (i.e., number of wires), is set by the designer at design time[and] [o]ne link (represented in Figure 11.1) defines the following signals… Pres.—

’449 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	Indicates the current priority of the packet used to define preferred traffic class (or Quality of Service). The width is fixed during the design time, allowing multiple pressure levels within the same NoC instance (bits 3–5 in Figure 11.2) [and] RxRdy—flow control”:  ***11.3.1.3 Physical Layer*** The delivery of packets within the NoC is the responsibility of the physical layer. Packets, which have been split by the transport layer into cells, are delivered as words that are sent along links. Within a single clock cycle, the physical layer may carry words comprising a fraction of a cell, a single cell, or multiple cells. The link size, or width (i.e., number of wires), is set by the designer at design time and determines the number of cells of one word. NTTP defines five possible link-widths: quarter (QRT), half (HLF), single (SGL), double (DBL), and quad (QUAD). A single-width (SGL) link transmits one cell per clock cycle, a double-width link transmits two cells per clock cycle, and so on. Words travel within point-to-point links, which are independent from other protocol layers: a word is sent through a transmit port, Tx, over a link to a receive port, Rx. The actual number of wires in a link depends on the

	maximum cell-width (header, necker, and data cell) and the link-width. One link (represented in Figure 11.1) defines the following signals: • **Data**—Data word of the width specified at design-time. • **Frm**—When asserted high, indicates that a packet is being transmitted. • **Head**—When asserted high, indicates the current word contains a packet header. When the link-width is smaller than single (SGL), the header transmission is split into several word transfers. However, the Head signal is asserted during the first transfer only. • **TailOfs**—Packet tail: when asserted high, indicates that the current word contains the last packet cell. When the link-width is smaller than single (SGL), the last cell transmission is split into several word transfers. However, the Tail signal is asserted during the first transfer only. • **Pres.**—Indicates the current priority of the packet used to define preferred traffic class (or Quality of Service). The width is fixed during the design time, allowing multiple pressure levels within the same NoC instance (bits 3–5 in Figure 11.2). • **Vld**—Data valid: when asserted high, indicates that a word is being transmitted. • **RxRdy**—Flow control: when asserted high, the receiver is ready to accept word. When de-asserted, the receiver is busy. This signal set, which constitutes the Media Independent NoC Interface (MINI), is the foundation for NTTP communications.

'449 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	See Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 313-314. As a further example, in the Arteris NoC, "QoS is achieved by exploiting the signal pressure embedded into the NTTP packet definition" where the "pressure signal can be generated by the IP itself and is typically linked to a certain level of urgency with which the transaction will have to be completed" and the "pressure information will be embedded in the NTTP packet at the NIU level":  **Quality of Service (QoS).** The QoS is a very important feature in the interconnect infrastructures because it provides a regulation mechanism allowing specification of guarantees on some of the parameters related to the traffic. Usually the end users are looking for guarantees on bandwidth and/or end-to-end communication latency. Different mechanisms and strategies have been proposed in the literature. For instance, in Æthereal NoC [11,24] proposed by NXP, a TDMA approach allows the specification of two traffic categories [25]: BE and GT. In the Arteris NoC, the QoS is achieved by exploiting the signal pressure embedded into the NTTP packet definition (Figures 11.1 and 11.2). The pressure

'449 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 signal can be generated by the IP itself and is typically linked to a certain level of urgency with which the transaction will have to be completed. For example, we can imagine associating the generation of the pressure signal when a certain threshold has been reached in the FIFO of the corresponding IP. This pressure information will be embedded in the NTTP packet at the NIU level: packets that have pressure bits equal to zero will be considered without QoS; packets with a nonzero value of the pressure bit will indicate preferred traffic class.* Such a QoS mechanism offers immediate service to the most urgent inputs and variables, and fair service whenever there are multiple contending inputs of equal urgency (BE). Within switches, arbitration decisions favor preferred packets and allocate remaining bandwidth (after preferred packets are served) fairly to contending packets. When there are contending preferred packets at the same pressure level, arbitration decisions among them are also fair. The Arteris NoC supports the following four different traffic classes:

'449 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 • **Real time and low latency (RTLL)**—Traffic flows that require the lowest possible latency. Sometimes it is acceptable to have brief intervals of longer latency as long as the average latency is low. Care must be taken to avoid starving other traffic flows as a side effect of pursuing low latency. • **Guaranteed throughput (GT)**—Traffic flows that must maintain their throughput over a relatively long time interval. The actual bandwidth needed can be highly variable even over long intervals. Dynamic pressure is employed for this traffic class. • **Guaranteed bandwidth (GBW)**—Traffic flows that require a guaranteed amount of bandwidth over a relatively long time interval. Over short periods, the network may lag or lead in providing this bandwidth. Bandwidth meters may be inserted onto links in the NoC to regulate these flows, using either of the two methods. If the flow is assigned high pressure, the meter asserts backpressure (flow control) to prevent the flow from exceeding a maximum bandwidth. Alternatively, the meter can modulate the flows pressure (priority) dynamically as needed to maintain an average bandwidth. • **Best effort (BE)**—Traffic flows that do not require guaranteed latency or throughput but have an expectation of fairness.

<table>
<tr><th>'449 Patent Claim</th><th>Motorola Product Including Snapdragon System on Chip[1]</th></tr>
<tr><td></td><td>

* Note that in the NTTP packet, the pressure field allows more then one bit, resulting in multiple levels of preferred traffic.

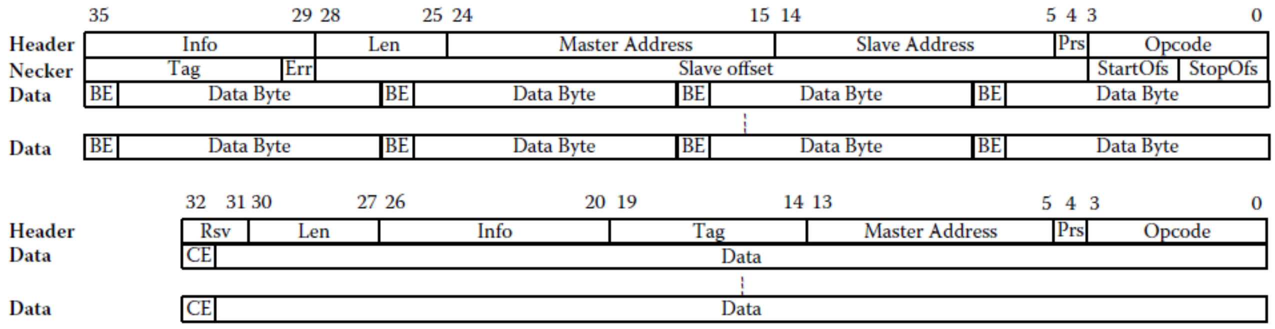
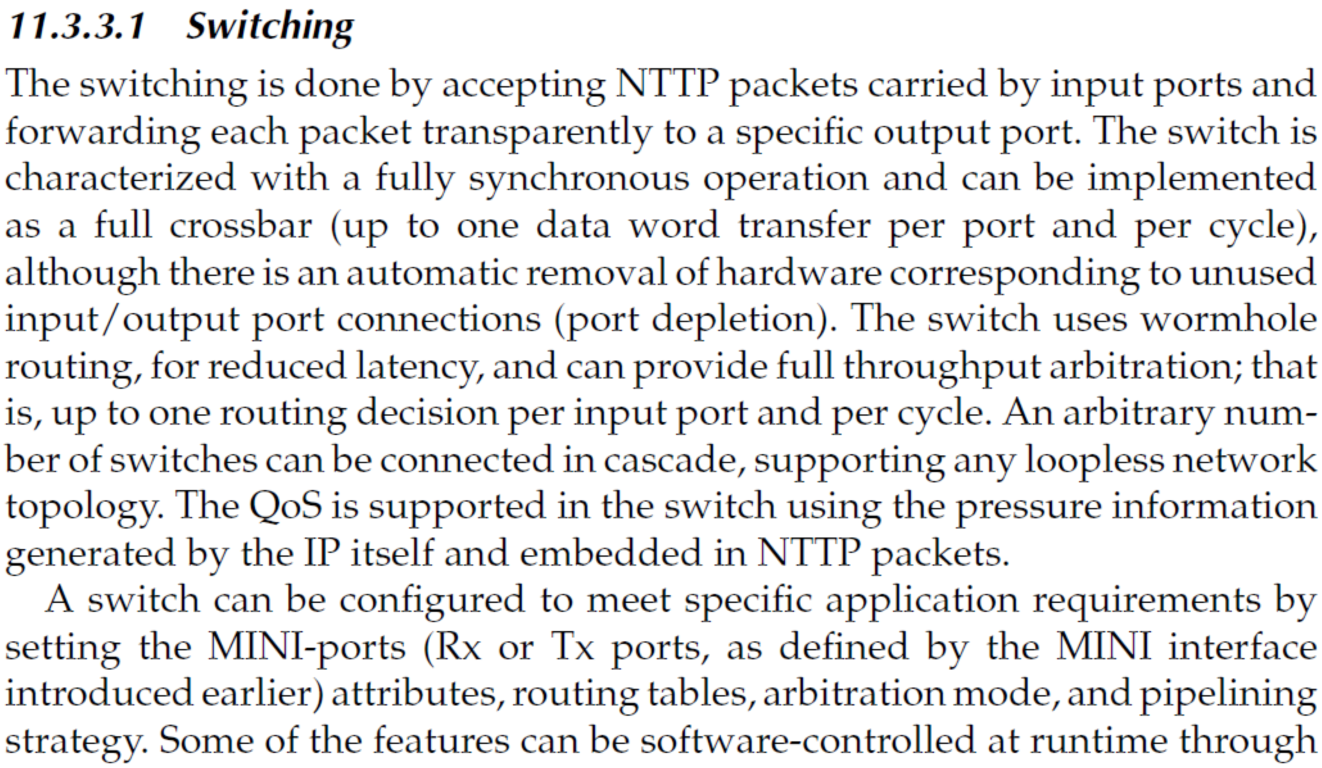


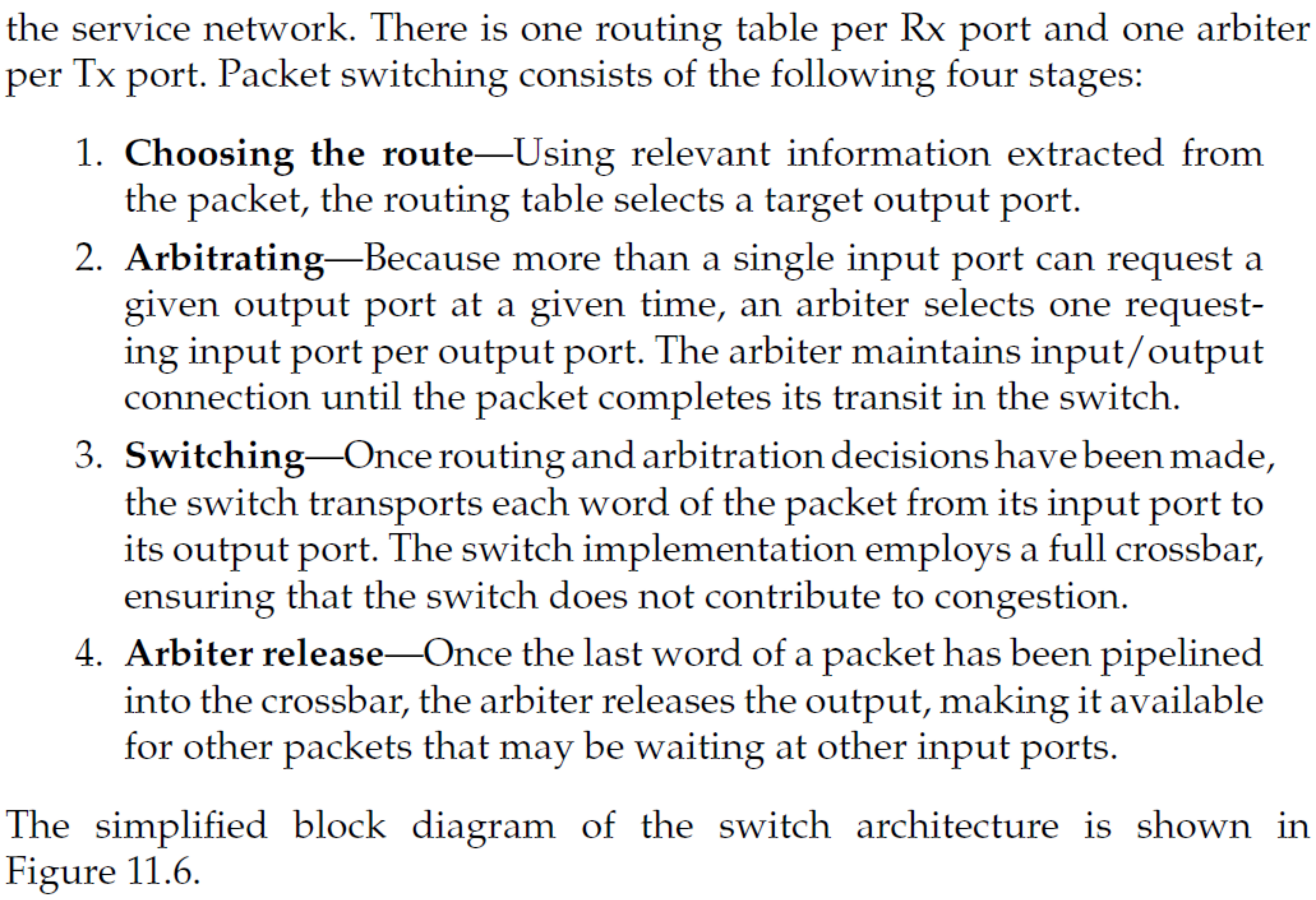
FIGURE 11.2
NTTP packet structure.

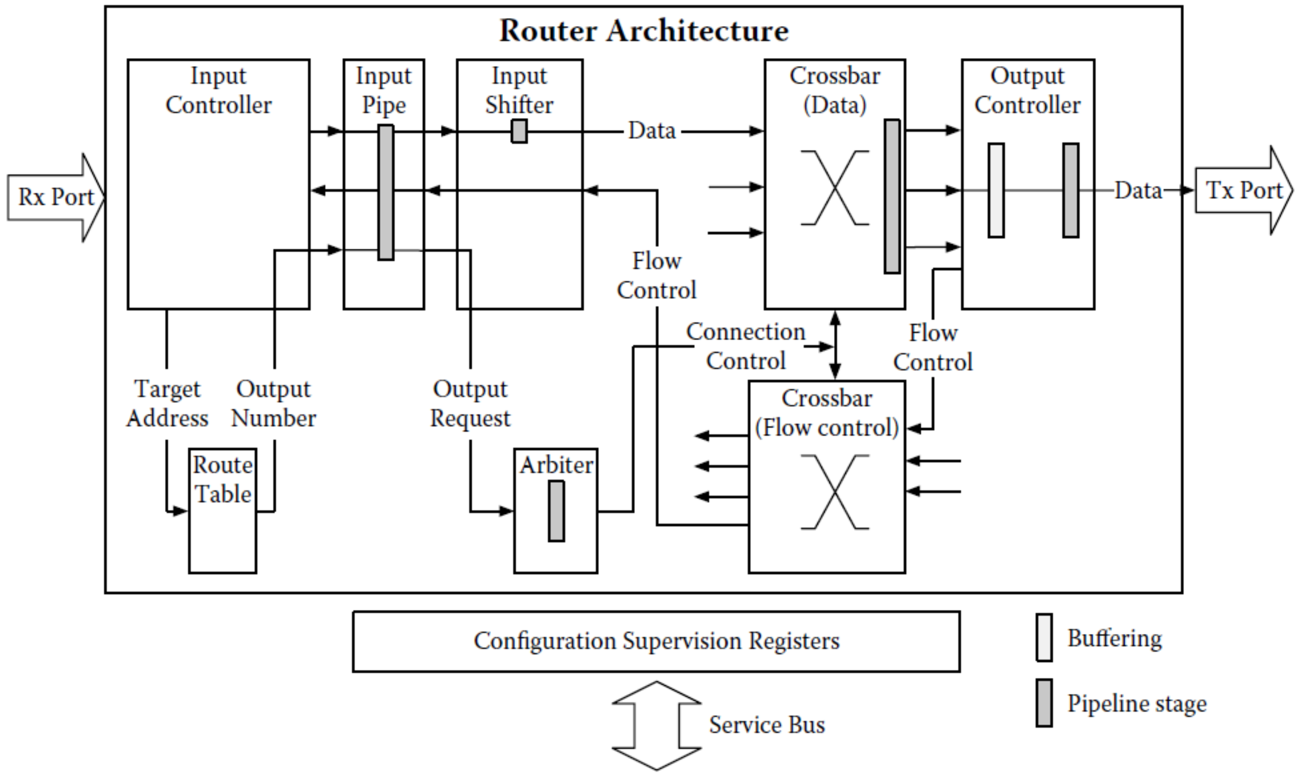
Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 313, 315-316.

As a further example, in the Arteris NoC, "QoS is supported in the switch using pressure information generated by the IP itself and embedded in NTTP packets" and the router architecture includes blocks such as "Input Controller," "Flow Control," "Crossbar (Flow control)" "Route Table" and "Arbiter":

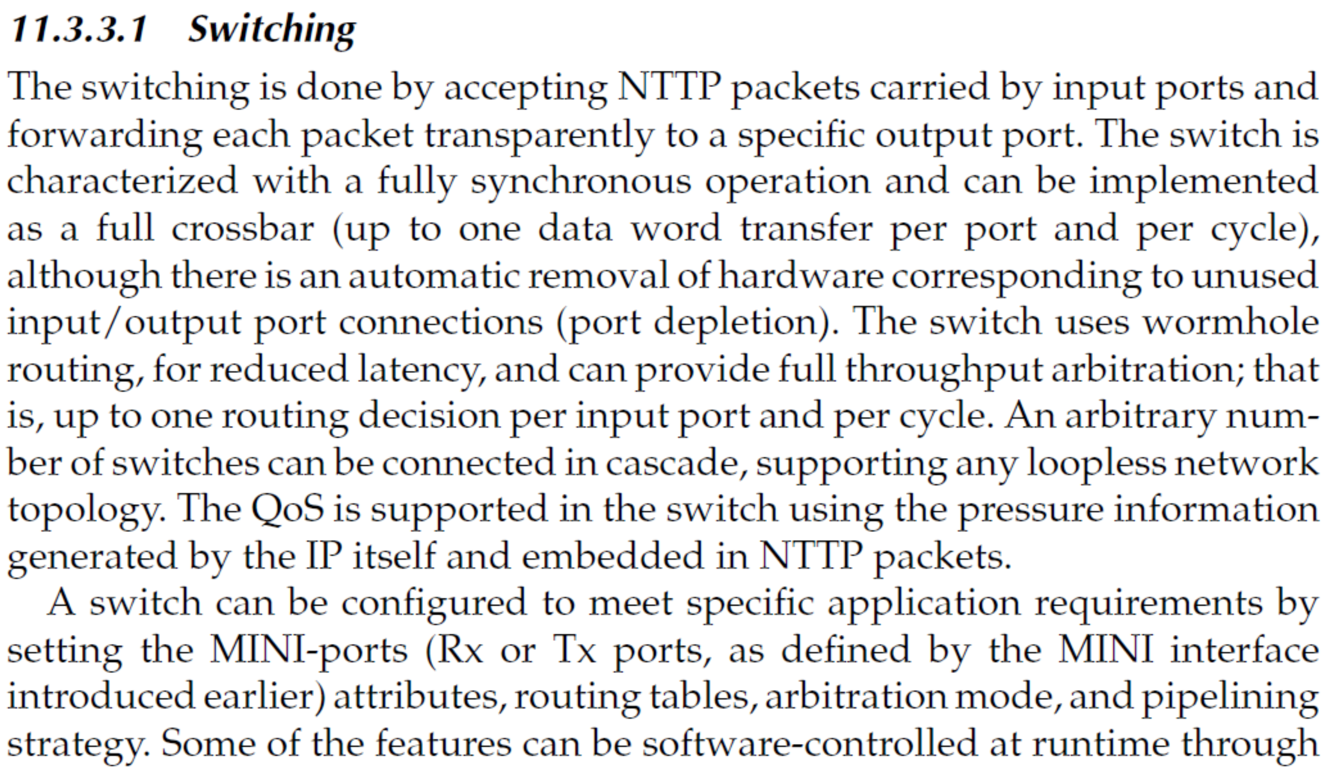
</td></tr>
</table>

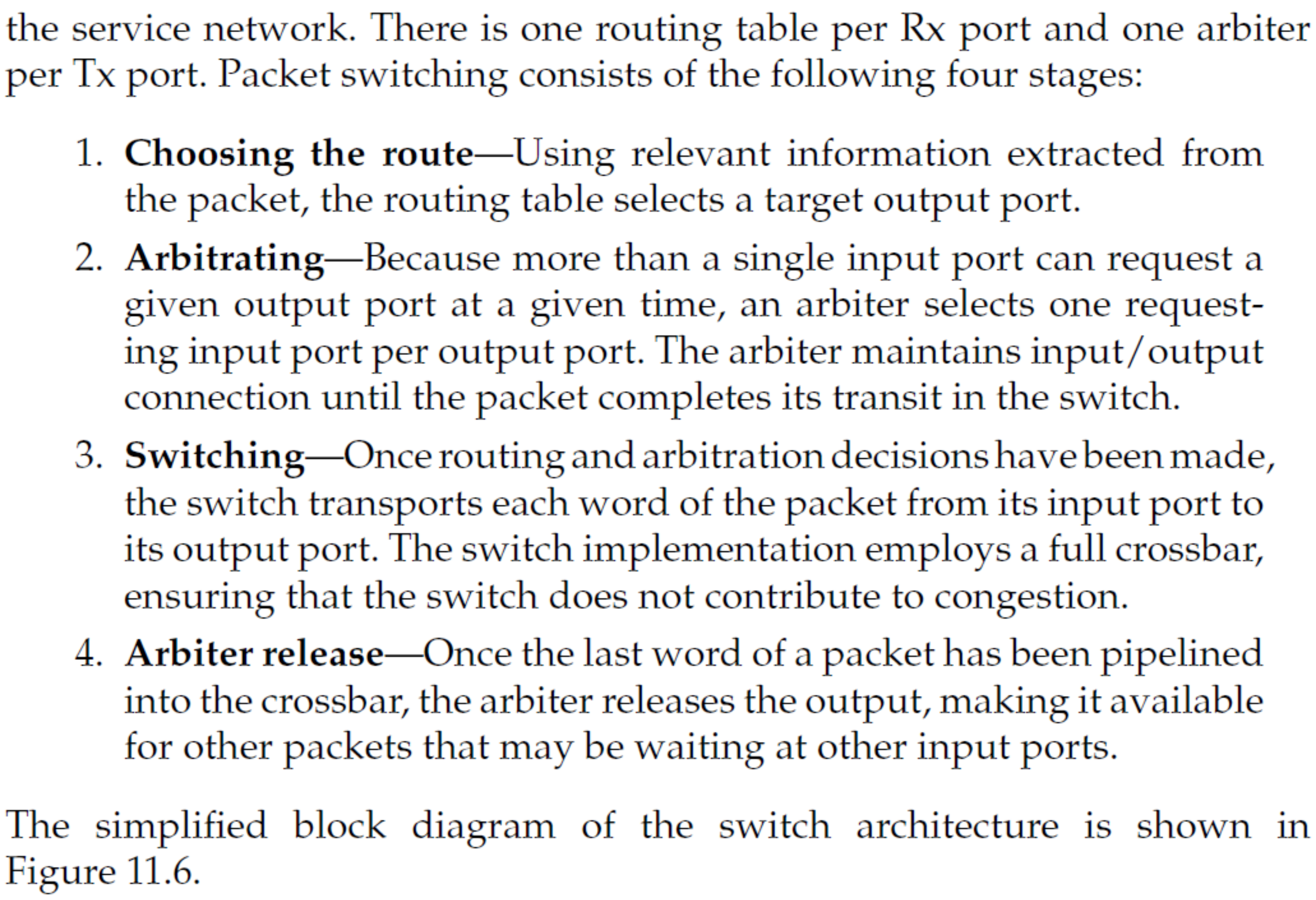
'449 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 ***11.3.3.1 Switching*** The switching is done by accepting NTTP packets carried by input ports and forwarding each packet transparently to a specific output port. The switch is characterized with a fully synchronous operation and can be implemented as a full crossbar (up to one data word transfer per port and per cycle), although there is an automatic removal of hardware corresponding to unused input/output port connections (port depletion). The switch uses wormhole routing, for reduced latency, and can provide full throughput arbitration; that is, up to one routing decision per input port and per cycle. An arbitrary number of switches can be connected in cascade, supporting any loopless network topology. The QoS is supported in the switch using the pressure information generated by the IP itself and embedded in NTTP packets. A switch can be configured to meet specific application requirements by setting the MINI-ports (Rx or Tx ports, as defined by the MINI interface introduced earlier) attributes, routing tables, arbitration mode, and pipelining strategy. Some of the features can be software-controlled at runtime through

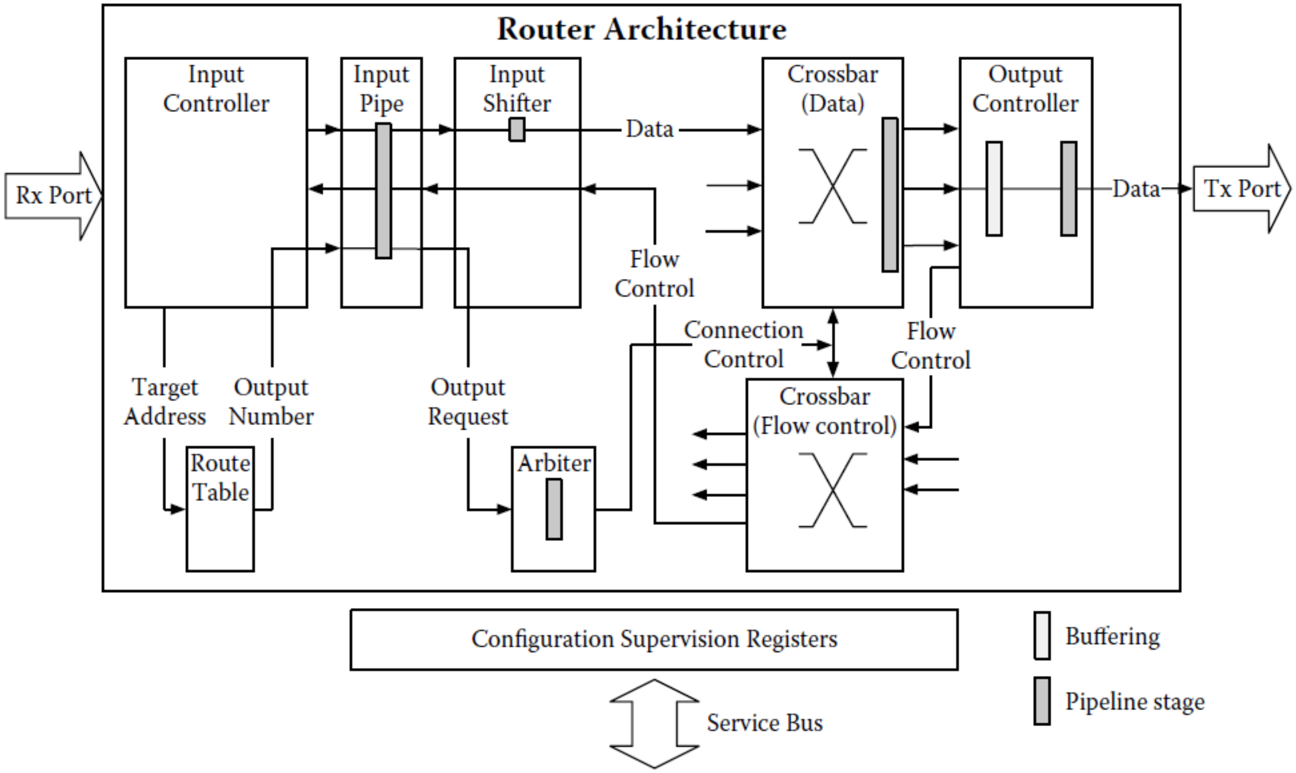
'449 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 the service network. There is one routing table per Rx port and one arbiter per Tx port. Packet switching consists of the following four stages: 1. **Choosing the route**—Using relevant information extracted from the packet, the routing table selects a target output port. 2. **Arbitrating**—Because more than a single input port can request a given output port at a given time, an arbiter selects one requesting input port per output port. The arbiter maintains input/output connection until the packet completes its transit in the switch. 3. **Switching**—Once routing and arbitration decisions have been made, the switch transports each word of the packet from its input port to its output port. The switch implementation employs a full crossbar, ensuring that the switch does not contribute to congestion. 4. **Arbiter release**—Once the last word of a packet has been pipelined into the crossbar, the arbiter releases the output, making it available for other packets that may be waiting at other input ports. The simplified block diagram of the switch architecture is shown in Figure 11.6.

'449 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 Router Architecture Rx Port Input Controller Input Pipe Input Shifter Data Crossbar (Data) Output Controller Data Tx Port Flow Control Connection Control Flow Control Target Address Output Number Output Request Crossbar (Flow control) Route Table Arbiter Configuration Supervision Registers Service Bus Buffering Pipeline stage **FIGURE 11.6** Packet transportation unit: Router architecture. *See* Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 319-320.

'449 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
the communication manager forwarding the request to a resource manager; the resource manager determining whether a target connection with the desired connection properties is available;	In the Arteris NoC in the Snapdragon SoC included in the Motorola product, the communication manager forwards the request to a resource manager and the resource manager determines whether a target connection with the desired connection properties is available, either literally or under the doctrine of equivalents. For example, in the Arteris NoC used by Snapdragon SoC included in the Motorola product, "QoS is supported in the switch" which "choos[es] the route" using a "routing table"; "arbitrat[es]"; and "switch[es]" and the router architecture includes blocks such as "Input Controller," "Flow Control," "Crossbar (Flow control)" "Route Table" and "Arbiter":

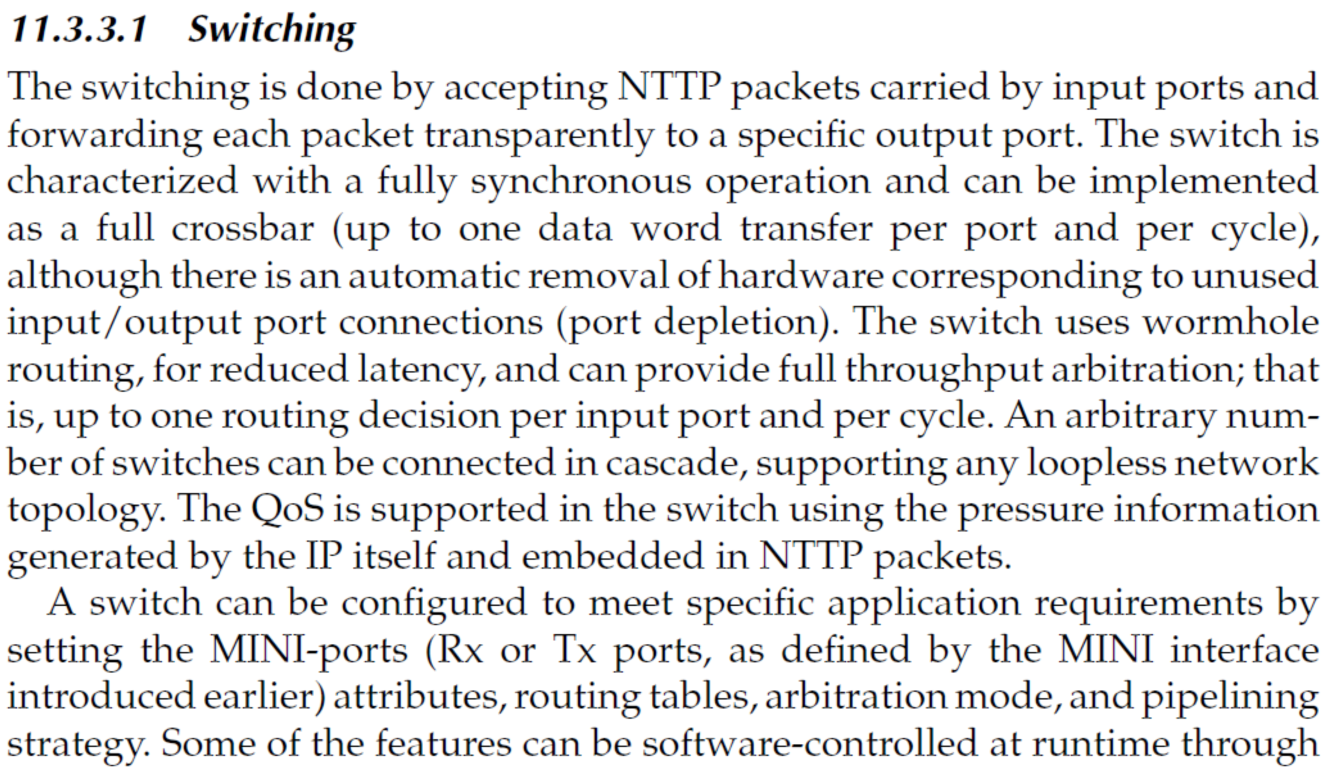
'449 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 ***11.3.3.1 Switching*** The switching is done by accepting NTTP packets carried by input ports and forwarding each packet transparently to a specific output port. The switch is characterized with a fully synchronous operation and can be implemented as a full crossbar (up to one data word transfer per port and per cycle), although there is an automatic removal of hardware corresponding to unused input/output port connections (port depletion). The switch uses wormhole routing, for reduced latency, and can provide full throughput arbitration; that is, up to one routing decision per input port and per cycle. An arbitrary number of switches can be connected in cascade, supporting any loopless network topology. The QoS is supported in the switch using the pressure information generated by the IP itself and embedded in NTTP packets. A switch can be configured to meet specific application requirements by setting the MINI-ports (Rx or Tx ports, as defined by the MINI interface introduced earlier) attributes, routing tables, arbitration mode, and pipelining strategy. Some of the features can be software-controlled at runtime through

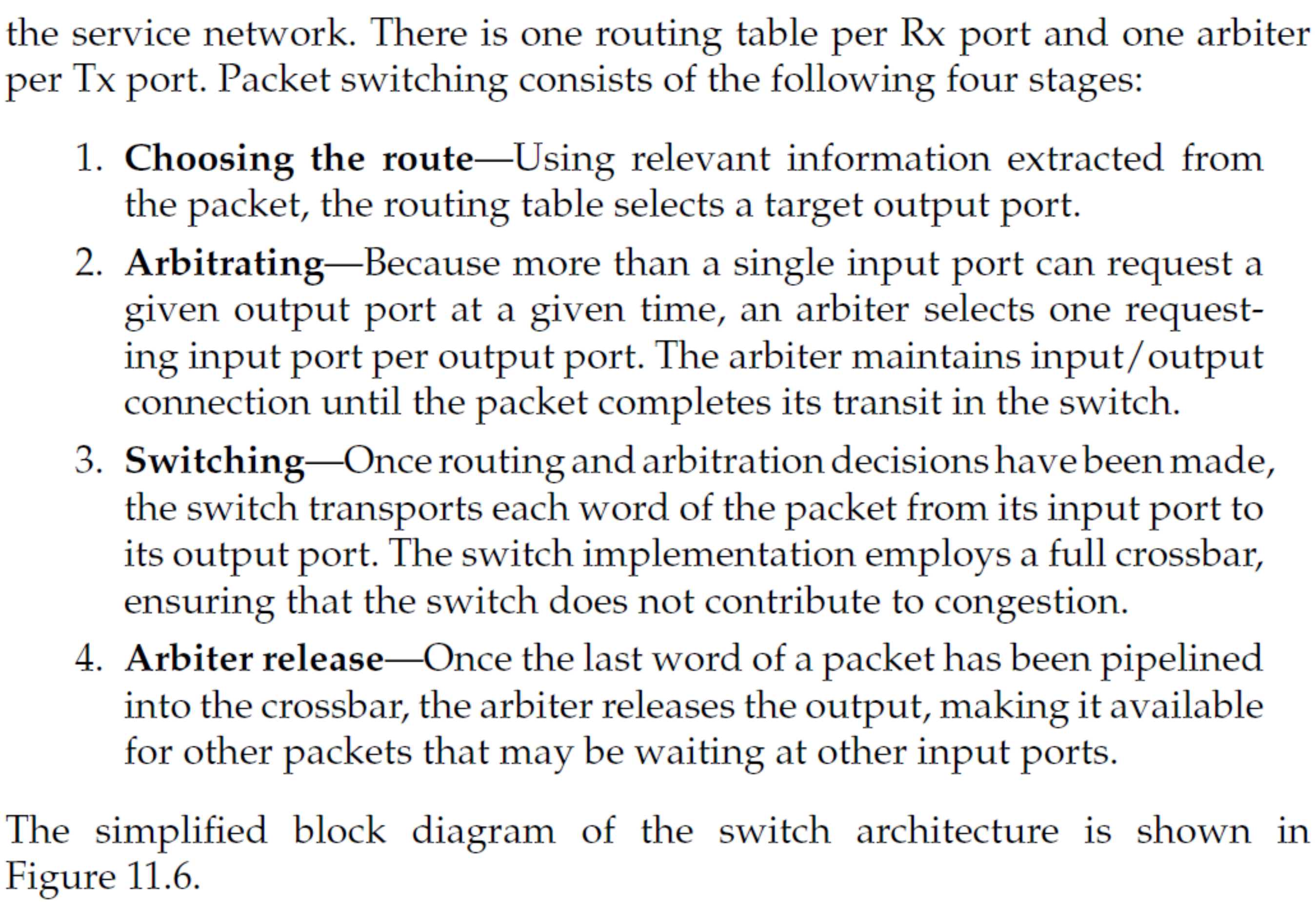
'449 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 the service network. There is one routing table per Rx port and one arbiter per Tx port. Packet switching consists of the following four stages: 1. **Choosing the route**—Using relevant information extracted from the packet, the routing table selects a target output port. 2. **Arbitrating**—Because more than a single input port can request a given output port at a given time, an arbiter selects one requesting input port per output port. The arbiter maintains input/output connection until the packet completes its transit in the switch. 3. **Switching**—Once routing and arbitration decisions have been made, the switch transports each word of the packet from its input port to its output port. The switch implementation employs a full crossbar, ensuring that the switch does not contribute to congestion. 4. **Arbiter release**—Once the last word of a packet has been pipelined into the crossbar, the arbiter releases the output, making it available for other packets that may be waiting at other input ports. The simplified block diagram of the switch architecture is shown in Figure 11.6.

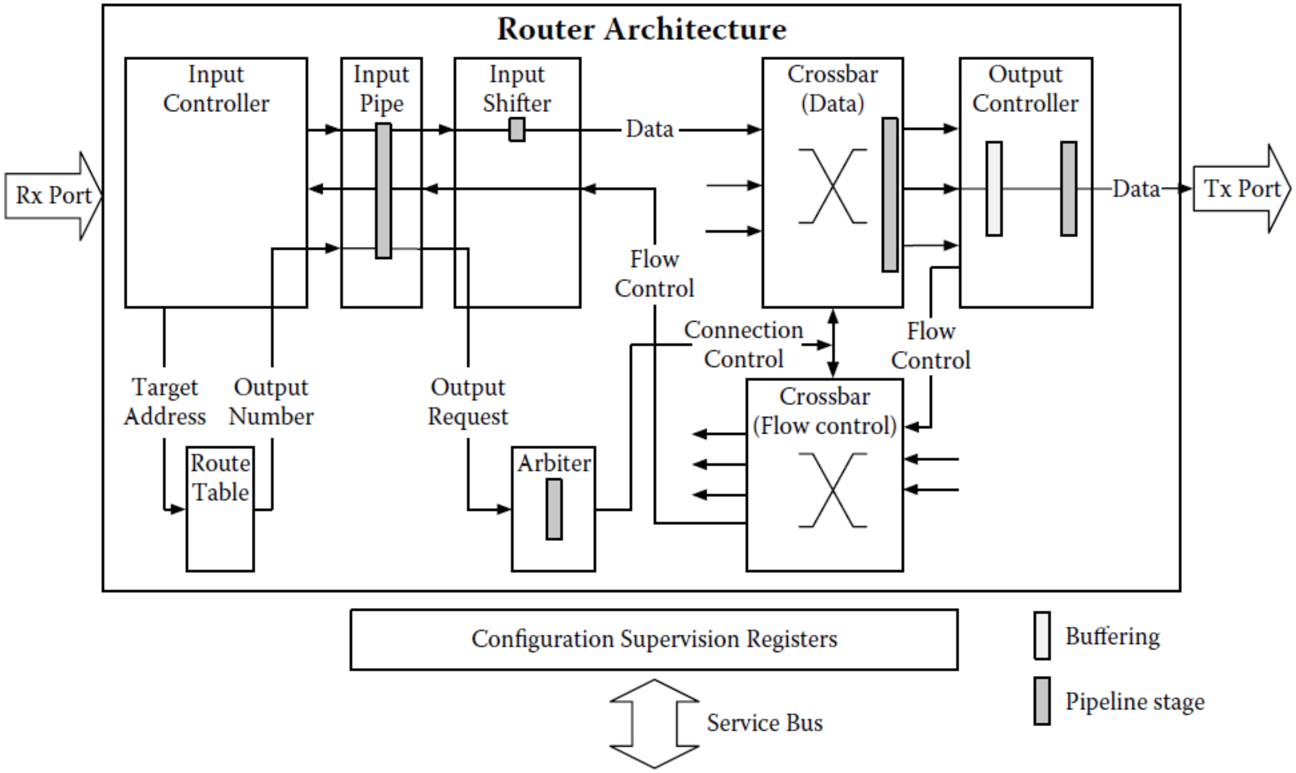
'449 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 Router Architecture Input Controller Input Pipe Input Shifter Data Crossbar (Data) Output Controller Rx Port Data Tx Port Flow Control Connection Control Flow Control Target Address Output Number Output Request Crossbar (Flow control) Route Table Arbiter Configuration Supervision Registers Service Bus Buffering Pipeline stage **FIGURE 11.6** Packet transportation unit: Router architecture. *See* Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 319-320.

'449 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	As a further illustration, in the Arteris NoC, "the pressure information used to define the preferred traffic class (QoS) of the requesting inputs… [t]he pressure information is given top priority by the switch arbiter" and "the input controller extracts pertinent data from packet headers, forwards it to the routing table, fetches back the target output number, and then sends a request to the arbiter. After arbitration is granted, the input controller transmits the rest of the packet to the crossbar. The request to the arbiter is sustained as long as the last word of the packet has not been transferred. Upon transferring the last cell of the packet, the arbiter is allowed to select a new input." *See* Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 321, 322.
the resource manager responding with the availability of the target connection to the communication manager; and	In the Arteris NoC in the Snapdragon SoC included in the Motorola product, the resource manager responds with the availability of the target connection to the communication manager, either literally or under the doctrine of equivalents. For example, in the Arteris NoC used by the Snapdragon SoC included in the Motorola product, "the pressure information used to define the preferred traffic class (QoS) of the requesting inputs… [t]he pressure information is given top priority by the switch arbiter" and "the input controller extracts pertinent data from packet headers, forwards it to the routing table, fetches back the target output number, and then sends a request to the arbiter. After arbitration is granted, the input controller transmits the rest of the packet to the crossbar. The request to the arbiter is sustained as long as the last word of the packet has not been transferred. Upon transferring the last cell of the packet, the arbiter is allowed to select a new input." *See* Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 321, 322.

'449 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	As a further illustration, in the Arteris NoC, "[t]he arbiter ensures that the connection matrix (a row per input and a column per output) contains at most one connection per column, that is, a given output is not fed by two inputs at the same time. The dual guarantee— at most one connection per row—is handled by the input controller. Each output has an arbiter that includes prefiltering. For maximum flexibility, each port can specify its own arbiter from the list of available arbiters (random, round robin, LRU, FIFO, or fixed priority)." *Id.* at 322-323.
the target connection between the first and second module being established based on the available properties of said communication channels of said connection.	In the Arteris NoC in the Snapdragon SoC included in the Motorola product, the target connection between the first and second module is established based on the available properties of said communication channels of said connection, either literally or under the doctrine of equivalents. For example, in the Arteris NoC used by the Snapdragon SoC included in the Motorola product, "QoS is supported in the switch" which "choos[es] the route" using a "routing table"; "arbitrat[es]"; and "switch[es]":

'449 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 *11.3.3.1 Switching* The switching is done by accepting NTTP packets carried by input ports and forwarding each packet transparently to a specific output port. The switch is characterized with a fully synchronous operation and can be implemented as a full crossbar (up to one data word transfer per port and per cycle), although there is an automatic removal of hardware corresponding to unused input/output port connections (port depletion). The switch uses wormhole routing, for reduced latency, and can provide full throughput arbitration; that is, up to one routing decision per input port and per cycle. An arbitrary number of switches can be connected in cascade, supporting any loopless network topology. The QoS is supported in the switch using the pressure information generated by the IP itself and embedded in NTTP packets. A switch can be configured to meet specific application requirements by setting the MINI-ports (Rx or Tx ports, as defined by the MINI interface introduced earlier) attributes, routing tables, arbitration mode, and pipelining strategy. Some of the features can be software-controlled at runtime through

'449 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	 the service network. There is one routing table per Rx port and one arbiter per Tx port. Packet switching consists of the following four stages: 1. **Choosing the route**—Using relevant information extracted from the packet, the routing table selects a target output port. 2. **Arbitrating**—Because more than a single input port can request a given output port at a given time, an arbiter selects one requesting input port per output port. The arbiter maintains input/output connection until the packet completes its transit in the switch. 3. **Switching**—Once routing and arbitration decisions have been made, the switch transports each word of the packet from its input port to its output port. The switch implementation employs a full crossbar, ensuring that the switch does not contribute to congestion. 4. **Arbiter release**—Once the last word of a packet has been pipelined into the crossbar, the arbiter releases the output, making it available for other packets that may be waiting at other input ports. The simplified block diagram of the switch architecture is shown in Figure 11.6.

'449 Patent Claim	**Motorola Product Including Snapdragon System on Chip[1]**
	 Router Architecture Rx Port Input Controller Input Pipe Input Shifter Data Crossbar (Data) Output Controller Data Tx Port Flow Control Connection Control Flow Control Target Address Output Number Output Request Crossbar (Flow control) Route Table Arbiter Configuration Supervision Registers Service Bus Buffering Pipeline stage **FIGURE 11.6** Packet transportation unit: Router architecture. *See* Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 319-320.

'449 Patent Claim	Motorola Product Including Snapdragon System on Chip[1]
	In the Arteris NoC, "the pressure information used to define the preferred traffic class (QoS) of the requesting inputs… [t]he pressure information is given top priority by the switch arbiter" and "the input controller extracts pertinent data from packet headers, forwards it to the routing table, fetches back the target output number, and then sends a request to the arbiter. After arbitration is granted, the input controller transmits the rest of the packet to the crossbar. The request to the arbiter is sustained as long as the last word of the packet has not been transferred. Upon transferring the last cell of the packet, the arbiter is allowed to select a new input." *See* Networks-On-Chips Theory and Practice, https://vdoc.pub/download/networks-on-chips-theory-and-practice-embedded-multi-core-systems-6f26qivv11f0, at 321, 322. As a further illustration, in the Arteris NoC, "[t]he arbiter ensures that the connection matrix (a row per input and a column per output) contains at most one connection per column, that is, a given output is not fed by two inputs at the same time. The dual guarantee— at most one connection per row—is handled by the input controller. Each output has an arbiter that includes prefiltering. For maximum flexibility, each port can specify its own arbiter from the list of available arbiters (random, round robin, LRU, FIFO, or fixed priority)." *Id.* at 322-323. As a further illustration, in the Arteris NoC, "[t]he crossbar implements datapath connection between inputs and outputs. It uses the connection matrix produced by the arbiter to determine which connections must be established. It is equivalent to a set of *m* muxes (one per output port), each having *n* inputs (one per input port)." *Id.* at 323.